Quadratic Metric Elicitation for Fairness and Beyond

Abstract

Metric elicitation is a recent framework for eliciting classification performance metrics that best reflect implicit user preferences based on the task and context. However, available elicitation strategies have been limited to linear (or quasi-linear) functions of predictive rates, which can be practically restrictive for many applications including fairness. This paper develops a strategy for eliciting more flexible multiclass metrics defined by quadratic functions of rates, designed to reflect human preferences better. We show its application in eliciting quadratic violation-based group-fair metrics. Our strategy requires only relative preference feedback, is robust to noise, and achieves near-optimal query complexity. We further extend this strategy to eliciting polynomial metrics – thus broadening the use cases for metric elicitation.

1 INTRODUCTION

Given a classification task, which performance metric should the classifier optimize? This question is often faced by practitioners while developing machine learning solutions. For example, consider cancer diagnosis where a doctor applies a cost-sensitive predictive model to classify patients into cancer categories [Yang and Naiman, 2014]. Although it is clear that the chosen costs directly determine the model decisions and dictate patient outcomes, it is not clear how to quantify the expert's intuition into precise quantitative cost trade-offs, i.e., the performance metric. Indeed this is also true for a variety of other domains including *fair machine learning* where picking the right metric is a critical challenge [Dmitriev and Wu, 2016, Zhang et al., 2020].

Hiranandani et al. [2019a, 2019b] addressed this issue by formalizing the framework of *Metric Elicitation (ME)*, whose goal is to estimate a performance metric using preference feedback from a user. The motivation is that by employing

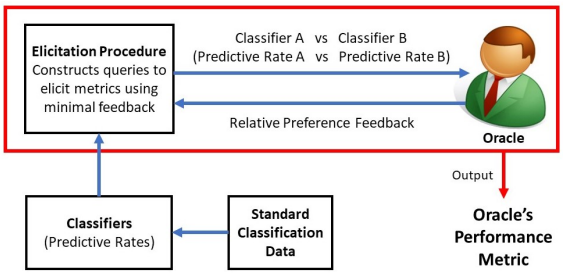


Figure 1: Metric Elicitation [Hiranandani et al., 2019a].

metrics that reflect a user's innate trade-offs given the task, context, and population at hand, one can learn models that best capture the user preferences [Hiranandani et al., 2019a]. As humans are often inaccurate in providing absolute quality feedback [Qian et al., 2013], Hiranandani et al. [2019a] propose to use pairwise comparison queries, where the user (oracle) is asked to compare two classifiers and provide a relative preference. Using such pairwise comparison queries, ME aims to recover the oracle's metric. Figure 1 (reproduced from Hiranandani et al. [2019a]) depicts the ME framework.

However, existing ME strategies only handle linear or quasi-linear function of predictive rates, which can be restrictive for many applications where the metrics are non-linear. For example, in *fair machine learning*, classifiers are often judged by measuring discrepancies between predictive rates for different protected groups [Hardt et al., 2016]. Similar discrepancy-based measures are also used in *distribution matching* applications [Narasimhan, 2018, Esuli and Sebastiani, 2015]. A common measure of discrepancy in such applications is the squared difference, which is a quadratic metric that cannot be handled by existing approaches. Similar quadratic metrics also find use in class-imbalanced learning [Goh et al., 2016, Narasimhan, 2018] (see Section 2.3 for examples). Motivated by these examples, in this paper, we propose strategies for eliciting metrics defined by *quadratic* functions of rates, that encompass linear metrics as special cases. We further extend our approach to elicit polynomial metrics, a universal family of functions [Stone, 1948], allowing one to better capture real-world human preferences.

Submitted to the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022). **To be used for reviewing only**.

Our high-level idea is to approximate the quadratic metric with multiple linear functions, employ linear ME to estimate the local slopes, and combine the slope estimates to reconstruct the original metric. While natural and elegant, this approach comes with non-trivial challenges. Firstly, we must choose center points for the local-linear approximations, and the chosen points must represent feasible queries. Secondly, because of the use of pairwise queries, we only receive *slopes* (directions) and not magnitudes for the local-linear functions, requiring intricate analyses to reconstruct the original metric and to deal with multiplicative errors that result. Despite the challenges, our method requires a query complexity that is only *linear* in the number of unknowns, which we show is *near-optimal*. To our knowledge, we are the first to prove such a lower bound for metric elicitation.

One important and timely application of our proposal is *eliciting fairness metrics* [Dwork et al., 2012, Kleinberg et al., 2017]. While several group-based fairness metrics have been proposed to capture bias in automated decision-making, selecting the right metric remains a crucial challenge [Zhang et al., 2020]. Recently, Hiranandani et al. [2020] proposed an approach for eliciting group-fair metrics that measure discrepancies using the absolute differences in rates across multiple sensitive groups. Unfortunately, their approach specifically handles metrics that are linear in the group discrepancies and does not generalize to other families of metrics. We extend this setup to allow for more general fairness metrics defined by quadratic functions of group discrepancies and show how our proposed quadratic ME approach can be easily adapted to elicit such group-fair quadratic metrics.

Our contributions are summarized as follows:

- We propose a novel quadratic metric elicitation algorithm for classification problems, which requires only pairwise preference feedback either over classifiers or predictive rates (Section 3).
- Specific to group-based fairness tasks, we show how to jointly elicit the predictive performance and fairness metrics, and the trade-off between them (Section 4).
- We show that the proposed approach is robust under feedback and finite sample noise and requires a near-optimal number of queries (Section 5).
- We empirically validate the proposal for multiple classes and groups on simulated oracles (Section 6).
- Lastly, we generalize our strategy to elicit higher-order polynomials by recursively applying the procedure to elicit lower-order approximations (Appendix H).

Notation. For $k \in \mathbb{Z}_+$, we denote $[k] = \{1, \cdots, k\}$ and use Δ_k to denote the $(k-1)$-dimensional simplex. We denote inner products by $\langle \cdot, \cdot \rangle$ and Hadamard products by $\odot$. $\|\cdot\|_F$ represents the Frobenius norm, and $\boldsymbol{\alpha}_i \in \mathbb{R}^q$ denotes the i-th standard basis vector, where the i-th coordinate is 1 and others are 0.

2 BACKGROUND

We consider a k-class classification setting with $X \in \mathcal{X}$ and $Y \in [k]$ denoting the input and output random variables, respectively. We assume access to an n-sized sample $\{(\mathbf{x}, y)_i\}_{i=1}^n$ generated *iid* from a distribution $\mathbb{P}(X, Y)$. We work with randomized classifiers $h : \mathcal{X} \to \Delta_k$ that for any $\mathbf{x}$ gives a distribution $h(\mathbf{x})$ over the k classes and use $\mathcal{H} = \{h : \mathcal{X} \to \Delta_k\}$ to denote the set of all classifiers.

Predictive rates: We denote the predictive rates for a classifier h by the vector $\mathbf{r}(h, \mathbb{P}) \in \mathbb{R}^k$, where the i-th coordinate is the fraction of label-i examples for which the randomized classifier h also predicts class i:

$$r_i(h, \mathbb{P}) := \mathbb{P}(h(X) = i | Y = i) \quad \text{for } i \in [k]. \tag{1}$$

The probability above is over draw of $(X, Y) \sim \mathbb{P}$ and the randomness in h. Our proposal easily extends to general predictive rates of the form $\mathbb{P}(h(X) = j | Y = i)$ for $i, j \in [k]$. For simplicity, we defer this extension to Appendix E.

Metrics: We consider metrics that are defined by a general function $\phi : [0, 1]^k \to \mathbb{R}$ of rates:

$$\phi(\mathbf{r}(h, \mathbb{P})).$$

This includes the (weighted) accuracy $\phi^{\text{acc}}(\mathbf{r}(h, \mathbb{P})) = \sum_i a_i r_i(h, \mathbb{P})$, for weights $a_i \in \mathbb{R}_+$, the G-mean, and many more metrics [Sokolova and Lapalme, 2009]. Unless specified, we treat metrics as utilities, i.e., larger values are better. Since the metric's scale does not affect the learning problem [Narasimhan et al., 2015], we allow $\phi : [0, 1]^k \to [0, 1]$.

Feasible rates: We will restrict our attention to only those rates that are feasible, i.e., can be achieved by some classifier. The set of all feasible rates is given by:

$$\mathcal{R} = \{\mathbf{r}(h, \mathbb{P}) \,:\, h \in \mathcal{H}\}.$$

To avoid clutter in notations, we will suppress the dependence on $\mathbb{P}$ and h if it is clear from the context.

2.1 METRIC ELICITATION: PROBLEM SETUP

We now describe the problem of *Metric Elicitation*, which follows from Hiranandani et al. [2019b]. There's an *unknown* metric ϕ, and we seek to elicit its form by posing queries to an *oracle* asking which of two classifiers is more preferred by it. The oracle has access to the metric ϕ and responds by comparing its value on the two classifiers.

Definition 1 (Oracle Query)**.** *Given two classifiers h_1, h_2 (equiv. to rates $\mathbf{r}_1, \mathbf{r}_2$ respectively), a query to the Oracle (with metric ϕ) is represented by:*

$$\Gamma(h_1, h_2 \,;\, \phi) = \Omega(\mathbf{r}_1, \mathbf{r}_2 \,;\, \phi) = \mathbb{1}[\phi(\mathbf{r}_1) > \phi(\mathbf{r}_2)], \tag{2}$$

where $\Gamma : \mathcal{H} \times \mathcal{H} \to \{0, 1\}$ and $\Omega : \mathcal{R} \times \mathcal{R} \to \{0, 1\}$. The query asks whether h_1 is preferred to h_2 (equiv. if $\mathbf{r}_1$ is preferred to $\mathbf{r}_2$), as measured by ϕ.

In practice, the oracle can be an expert, a group of experts, or an entire user population. The ME framework can be applied by posing classifier comparisons directly via interpretable learning techniques [Ribeiro et al., 2016] or via A/B testing [Tamburrelli and Margara, 2014]. For example, in an internet-based application one may perform the A/B test by deploying two classifiers A and B with two different sub-populations of users and use their level of engagement to decide the preference over the two classifiers. For other applications, one may present visualizations of rates of the two classifiers (e.g., [Shen et al., 2020]), and have the user provide the preference (see Appendix K for an example). Moreover, since the metrics we consider are functions of only the predictive rates, queries comparing classifiers are the same as queries on the associated rates. So for convenience, we will have our algorithms pose queries comparing two (feasible) rates. Indeed given a feasible rate, one can efficiently find the associated classifier (see Appendix B.1 for details). We next formally state the ME problem.

Definition 2 (Metric Elicitation with Pairwise Queries (given $\{(\mathbf{x}, y)_i\}_{i=1}^n$) [Hiranandani et al., 2019a,b]). *Suppose that the oracle's (unknown) performance metric is ϕ. Using oracle queries of the form $\Omega(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2\,;\,\phi)$, where $\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2$ are the estimated rates from samples, recover a metric $\hat{\phi}$ such that $\|\phi - \hat{\phi}\| < \kappa$ under a suitable norm $\|\cdot\|$ for sufficiently small error tolerance $\kappa > 0$.*

The performance of ME is evaluated both by the query complexity and the quality of the elicited metric [Hiranandani et al., 2019a,b]. As is standard in the decision theory literature [Koyejo et al., 2015], we present our ME approach by first assuming access to population quantities such as the population rates $\mathbf{r}(h, \mathbb{P})$, then examine estimation error from finite samples, i.e., with empirical rates $\hat{\mathbf{r}}(h, \{(\mathbf{x}, y)_i\}_{i=1}^n)$.

2.2 LINEAR METRIC ELICITATION

As a warm up, we overview the Linear Performance Metric Elicitation (LPME) procedure of [Hiranandani et al., 2019b], which we will use as a subroutine. Here we assume that the oracle's metric is a linear function of rates $\phi^{\text{lin}}(\mathbf{r}) := \langle \mathbf{a}, \mathbf{r} \rangle$, for some unknown weights $\mathbf{a} \in \mathbb{R}^k$. In other words, given two rates $\mathbf{r}_1$ and $\mathbf{r}_2$, the oracle returns $\mathbb{1}[\langle \mathbf{a}, \mathbf{r}_1 \rangle > \langle \mathbf{a}, \mathbf{r}_2 \rangle]$. Since the metrics are scale invariant [Narasimhan et al., 2015], without loss of generality (w.l.o.g.), one may assume $\|\mathbf{a}\|_2 = 1$. The goal is to elicit (the slope of) $\mathbf{a}$ using pairwise comparisons over rates.

When the number of classes $k = 2$, the coefficients $\mathbf{a}$ can be elicited using a one-dimensional binary search. When $k > 2$, one can apply a coordinate-wise procedure, performing a binary search in one coordinate, while keeping the others fixed. The efficacy of this procedure, however, hinges on the geometry of the set of rates $\mathcal{R}$. Before discussing the geometry, we make a mild assumption that ensures some signal for non-trivial classification.

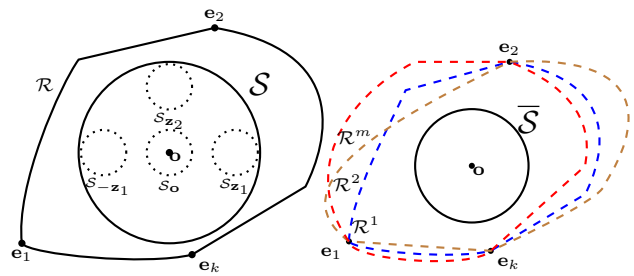


Figure 2: (a) Geometry of the set of predictive rates $\mathcal{R}$: A convex set enclosing a sphere $\mathcal{S}$ with trivial rates $\mathbf{e}_i \,\forall\, i \in [k]$ as vertices; (b) Geometry of the product set of group rates $\mathcal{R}^1 \times \cdots \times \mathcal{R}^m$ (best seen in color) [Hiranandani et al., 2020]; $\mathcal{R}^u \,\forall\, u \in [m]$ are convex sets with common vertices $\mathbf{e}_i \,\forall\, i \in [k]$ and enclose a sphere $\overline{\mathcal{S}} \subset \mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$.

Assumption 1. *The conditional-class distributions are distinct, i.e., $P(Y = i|X) \neq P(Y = j|X) \,\forall\, i \neq j$.*

Let $\mathbf{e}_i \in \{0, 1\}^k$ denote the rates achieved by a trivial classifier that predicts class i for all inputs.

Proposition 1 (Geometry of $\mathcal{R}$; Figure 2(a)). *The set of rates $\mathcal{R} \subseteq [0, 1]^k$ is convex, has vertices $\{\mathbf{e}_i\}_{i=1}^k$, and contains the rate profile $\mathbf{o} = \frac{1}{k}\sum_{i=1}^k \mathbf{e}_i$ in the interior. Moreover, $\mathbf{o}$ is achieved by the uniform random classifier which for any input predicts each class with equal probability.*

Remark 1 (Existence of sphere $\mathcal{S}$). *Since $\mathcal{R}$ is convex and contains the point $\mathbf{o}$ in the interior, there exists a sphere $\mathcal{S} \subset \mathcal{R}$ of non-zero radius ρ centered at $\mathbf{o}$.*

By restricting the coordinate-wise binary search procedure to posing queries from within a sphere, LPME can be seen as minimizing a strongly-convex function and shown to converge to a solution $\hat{\mathbf{a}}$ close to $\mathbf{a}$. Specifically, the LPME procedure takes any sphere $\mathcal{S} \subset \mathcal{R}$, binary-search tolerance ϵ, and the oracle Ω (with metric ϕ^{lin}) as input, and by posing $O(k \log(1/\epsilon))$ queries recovers coefficients $\hat{\mathbf{a}}$ with $\|\mathbf{a} - \hat{\mathbf{a}}\|_2 \leq O(\sqrt{k}\epsilon)$. The details are provided in Appendix A.

Remark 2 (LPME Guarantee). *Given any k-dimensional sphere $\mathcal{S} \subset \mathcal{R}$ and an oracle Ω with metric $\phi^{\text{lin}}(\mathbf{r}) := \langle \mathbf{a}, \mathbf{r} \rangle$, the LPME algorithm (Algorithm 2, Appendix A) provides an estimate $\hat{\mathbf{a}}$ with $\|\hat{\mathbf{a}}\|_2 = 1$ such that the estimated slope is close to the true slope, i.e., ${}^{a_i}/_{a_j} \approx {}^{\hat{a}_i}/_{\hat{a}_j} \;\forall\, i, j \in [k]$.*

Note that the LPME procedure is closely tied to the scale invariance condition and thus only estimates the slope (direction) of the coefficient vector $\mathbf{a}$, and not its magnitude. Despite this drawback, we will discuss how we can elicit quadratic metrics using LPME in Section 3. Also note the algorithm takes as input an *arbitrary* sphere $\mathcal{S} \subset \mathcal{R}$, and restricts its queries to rate vectors within the sphere. In Appendix B.1, we discuss an efficient procedure [Hiranandani et al., 2019b] for identifying a sphere of suitable radius.

2.3 QUADRATIC PERFORMANCE METRICS

Equipped with the LPME subroutine, our aim is to elicit metrics that are quadratic functions of rates.

Definition 3 (Quadratic Metric)**.** *For a vector* $\mathbf{a} \in \mathbb{R}^k$ *and a negative semi-definite matrix* $\mathbf{B} \in NSD_k$ *with* $\|\mathbf{a}\|_2^2 + \|\mathbf{B}\|_F^2 = 1$ *(w.l.o.g. due to scale invariance):*

$$\phi^{\text{quad}}(\mathbf{r}\,;\,\mathbf{a},\mathbf{B}) = \langle \mathbf{a}, \mathbf{r}\rangle + \frac{1}{2}\mathbf{r}^T\mathbf{B}\mathbf{r}. \tag{3}$$

This family trivially includes the linear metrics as well as many modern metrics outlined below:

Example 1 (Class-imbalanced learning)**.** In problems with imbalanced class proportions, it is common to use metrics that emphasize equal performance across all classes. One example is Q-mean [Menon et al., 2013], which is the quadratic mean of rates: $\phi^{\text{qmean}}(\mathbf{r}) = 1 - 1/k \sum_{i=1}^{k} (1 - r_i)^2$.

Example 2 (Fairness violation)**.** A popular criterion for group-based fairness is equalized odds, which requires equal rates across different protected groups [Bechavod and Ligett, 2017]. This can be measured by the squared differences between the group rates. With m groups and $\mathbf{r}^g$ denoting the rate vector evaluated on examples from group g, this is given by: $\phi^{\text{EO}}((\mathbf{r}^1, \ldots, \mathbf{r}^m)) = 1 - [k\binom{m}{2}]^{-1} \sum_{v>u} \sum_{i=1}^{k} \left(r_i^u - r_i^v\right)^2$. Other quadratic fairness criteria for two classes include equal opportunity $\phi^{\text{EOpp}}((\mathbf{r}^1, \ldots, \mathbf{r}^m)) = 1 - \binom{m}{2}^{-1} \sum_{v>u} (r_1^u - r_1^v)^2$ [Hardt et al., 2016], balance for the negative class $\phi^{\text{BN}}((\mathbf{r}^1, \ldots, \mathbf{r}^m)) = 1 - \binom{m}{2}^{-1} \sum_{v>u} (r_2^u - r_2^v)^2$ [Kleinberg et al., 2017], error-rate balance $\phi^{\text{EB}}((\mathbf{r}^1, \ldots, \mathbf{r}^m)) = 1 - \binom{m}{2}^{-1} \frac{1}{2} \sum_{v>u} (r_1^u - r_1^v)^2 + (r_2^u - r_2^v)^2$ [Chouldechova, 2017], etc. and their weighted variants. In Section 4, we consider metrics that trade-off between an error term and a quadratic fairness violation term.

Example 3 (Distribution matching)**.** In certain binary classification applications, one needs the proportion of predictions for each class (i.e., the coverage) to match a target distribution $\boldsymbol{\pi} \in \Delta_2$ [Goh et al., 2016, Narasimhan, 2018]. A measure often used for this task is the squared difference between the per-class coverage and the target distribution: $\phi^{\text{cov}}(\mathbf{r}) = 1 - \frac{1}{2} \sum_{i=1}^{2} \left(\text{cov}_i(\mathbf{r}) - \pi_i\right)^2$, where $\text{cov}_i(\mathbf{r}) = r_i + 1 - r_{\neq i}$. Similar metrics can be found in the quantification literature where the target is set to the class prior $\mathbb{P}(Y = i)$ [Esuli and Sebastiani, 2015, Kar et al., 2016]. We capture more general quadratic distance measures for distributions, e.g. $(\text{cov}(\mathbf{r}) - \boldsymbol{\pi})^{\mathbf{T}}\mathbf{Q}(\text{cov}(\mathbf{r}) - \boldsymbol{\pi})$ for $\mathbf{Q} \in NSD_2$ [Lindsay et al., 2008].

Lastly, we need the following assumption on the metric.

Assumption 2. *The gradient of* ϕ *at the trivial rate* $\mathbf{o}$ *is non-zero, i.e.,* $\nabla\phi^{\text{quad}}(\mathbf{r})|_{\mathbf{r}=\mathbf{o}} = \mathbf{a} + \mathbf{B}\mathbf{o} \neq 0$.

The non-zero gradient assumption is reasonable for a concave ϕ^{quad}, where it merely implies that the optimal classifier for the metric is not the uniform random classifier.

3 QUADRATIC METRIC ELICITATION

We now present our procedure for Quadratic Performance Metric Elicitation (QPME). We assume that the oracle's unknown metric is quadratic (Definition 3) and seek to estimate its parameters $(\mathbf{a}, \mathbf{B})$ by posing queries to the oracle. Unlike LPME, a simple binary search based procedure cannot be directly applied to elicit these parameters. Our approach instead approximates the quadratic metric by a linear function at a few select but *feasible* rate vectors and invokes LPME to estimate the local-linear approximations' slopes. One of the key challenges is to pick a small number of *feasible* rates for performing the local approximations and to reconstruct the original metric *just* from the estimated local slopes.

3.1 LOCAL LINEAR APPROXIMATION

We will find it convenient to work with a shifted version of the quadratic metric, centered at the point $\mathbf{o}$, the uniform random rate vector (see Proposition 1):

$$\begin{aligned}\phi^{\text{quad}}(\mathbf{r};\,\mathbf{a},\mathbf{B}) &= \langle \mathbf{d}, \mathbf{r} - \mathbf{o}\rangle + \frac{1}{2}(\mathbf{r} - \mathbf{o})^T\mathbf{B}(\mathbf{r} - \mathbf{o}) + c \\ &= \bar{\phi}(\mathbf{r};\,\mathbf{d},\mathbf{B}) + c,\end{aligned} \tag{4}$$

where $\mathbf{d} = \mathbf{a} + \mathbf{B}\mathbf{o}$ and c is a constant independent of $\mathbf{r}$, and so the oracle can be equivalently seen as responding with the shifted metric $\bar{\phi}(\mathbf{r};\,\mathbf{d},\mathbf{B})$.

Note that, due to the scale invariance condition in Definition 3, the largest singular value of $\mathbf{B}$ is bounded by 1. This is because $\|\mathbf{B}\|_2 \leq \|\mathbf{B}\|_F \leq 1$. Thus the metric ϕ^{quad} is 1-smooth and implies that it is locally linear around a given rate. To this end, let z be a fixed point in $\mathcal{R}$, then the metric can be closely approximated by its first-order Taylor expansion in a small neighborhood around $\mathbf{z}$, for a constant c' as follows:

$$\bar{\phi}(\mathbf{r};\,\mathbf{d},\mathbf{B}) \approx \langle \mathbf{d} + \mathbf{B}(\mathbf{z} - \mathbf{o}), \mathbf{r}\rangle + c'. \tag{5}$$

So if we apply LPME to the metric $\bar{\phi}$ with the queries $(\mathbf{r}_1, \mathbf{r}_2)$ to the oracle restricted to a small ball around $\mathbf{z}$, the procedure effectively estimates the slope of the vector $\mathbf{d} + \mathbf{B}(\mathbf{z} - \mathbf{o})$ in the above linear function (up to a small approximation error).

We exploit this idea by applying LPME to small neighborhoods around selected points to elicit the coefficients $\mathbf{a}$ and $\mathbf{B}$ for the original metric in (3). For simplicity, we will assume that the oracle is noise-free and later show robustness to noise and the query complexity guarantees in Section 5.

3.2 ELICITING METRIC COEFFICIENTS

We outline the main steps of Algorithm 1 below. Please see Appendix C for the full derivation.

Estimate coefficients $\mathbf{d}$ (Line 1). We first wish to estimate the linear portion $\mathbf{d}$ of the metric $\bar{\phi}$ in (4). For this, we apply the LPME subroutine to a small ball $\mathcal{S}_{\mathbf{o}} \subset \mathcal{S}$ of

radius $\varrho < \rho$ around the point $\mathbf{o}$ (Fig. 2(a) illustrates this). Within this ball, the metric $\overline{\phi}$ approximately equals the linear function $\langle \mathbf{d}, \mathbf{r} \rangle + c'$ (see (5)), and so the LPME gives us an estimate of the slope of $\mathbf{d}$. From Remark 2, the estimates $\mathbf{f}_0 = (f_{10}, \ldots, f_{k0})$ approximately satisfy the following $(k-1)$ equations:

$$\frac{d_i}{d_1} = \frac{f_{i0}}{f_{10}} \qquad \forall\, i \in \{2, \ldots, k\}. \tag{6}$$

Estimate coefficients B (Lines 2–4). Next, we wish to estimate each column of the matrix $\mathbf{B}$ of the metric $\overline{\phi}$ in (4). For this, we apply LPME to small neighborhoods around points in the direction of standard basis vectors $\boldsymbol{\alpha}_j \in \mathbb{R}^k$, $j = 1, \ldots, k$. Note that within a small ball around $\mathbf{o} + \boldsymbol{\alpha}_j$, the metric $\overline{\phi}$ is approximately the linear function $\langle \mathbf{d} + \mathbf{B}_{:,j}, \mathbf{r} \rangle + c'$, and so the LPME procedure when applied to this region will give us an estimate of the slope of $\mathbf{d} + \mathbf{B}_{:,j}$. However, to ensure that the center point we choose is a feasible rate, we will have to re-scale the standard basis, and apply the subroutine to balls $\mathcal{S}_{\mathbf{z}_j}$ of radius $\varrho < \rho$ centered at $\mathbf{z}_j = \mathbf{o} + (\rho - \varrho)\boldsymbol{\alpha}_j$. See Figure 2(a) for the visual intuition. The returned estimates $\mathbf{f}_j = (f_{1j}, \ldots, f_{kj})$ approximately satisfy:

$$\frac{d_i + (\rho - \varrho)B_{ij}}{d_1 + (\rho - \varrho)B_{1j}} = \frac{f_{ij}}{f_{1j}} \quad \forall\, i \in \{2, \ldots, k\},\ j \le i. \tag{7}$$

Now note that since we are only eliciting slopes using LPME, we always lose out on one degree of freedom. However, the matrix $\mathbf{B}$ is symmetric, thus we have $k(k+1)/2 - 1$ equations. There are $k(k+1)/2 + k$ unknown entities in $\mathbf{a}$ and $\mathbf{B}$, and to estimate them we need 1 more equation besides the normalization condition. For this, we apply LPME to a sphere $\mathcal{S}_{-\mathbf{z}_1}$ of radius ϱ around rate $-\mathbf{z}_1$ as shown in Figure 2(a). The returned slopes $\mathbf{f}_1^- = (f_{11}^-, \ldots, f_{k1}^-)$ approximately satisfy:

$$\frac{d_2 - (\rho - \varrho)B_{21}}{d_1 - (\rho - \varrho)B_{11}} = \frac{f_{21}^-}{f_{11}^-}. \tag{8}$$

Putting it together (Line 5). By combining (6), (7) and (8), and denoting $F_{i,j,l} = f_{il}/f_{jl}$ and $F_{i,j,l}^- = f_{il}^-/f_{jl}^-$, we express each entry of $\mathbf{B}$ in terms of d_1 as follows:

$$B_{ij} = \Big(F_{i,1,j}(1 + F_{j,1,1}) - F_{i,1,j}F_{j,1,0}d_1 - F_{i,1,0} + F_{i,1,j}\frac{F_{2,1,1}^- + F_{2,1,1} - 2F_{2,1,0}}{F_{2,1,1}^- - F_{2,1,1}} \Big) d_1. \tag{9}$$

Using $\mathbf{d} = \mathbf{a} + \mathbf{B}\mathbf{o}$ and the fact that the coefficients are normalized, i.e., $\|\mathbf{a}\|_2^2 + \|\mathbf{B}\|_F^2 = 1$, we can obtain estimates for $\mathbf{B}$ and $\mathbf{a}$ independent of d_1. Note that the derivation so far assumes $d_1 \neq 0$. This is based on Assumption 2 that at least one coordinate of $\mathbf{d}$ is non-zero, which w.l.o.g. we take to be d_1. In practice, we can identify a non-zero coordinate using q trivial queries of the form $(\varrho\boldsymbol{\alpha}_i + \mathbf{o}, \mathbf{o}), \forall i \in [k]$.

Technical novelty. We emphasize that a key difference with Hiranandani et al. [2019a, 2019b] is that they rely on a boundary point characterization which may not hold for general nonlinear metrics. Instead, we use structural properties of the metric to estimate local-linear approximations. While this may be a convenient approach (given LPME), as discussed in Section 1, implementing it involves non-trivial challenges, such as: (a) working with *only* slopes for the local-linear functions, (b) ensuring that the center points for the approximations are feasible, and (c) handling multiplicative errors in the analysis (see Section 5).

Algorithm 1: QPM Elicitation

Input: $\mathcal{S}$, Search tolerance $\epsilon > 0$, Oracle Ω with metric $\overline{\phi}$

1: $\mathbf{f}_0 \leftarrow$ LPME$(\mathcal{S}_{\mathbf{o}}, \epsilon, \Omega)$ with $\mathcal{S}_{\mathbf{o}} \subset \mathcal{S}$ and obtain (6)

2: **For** $j \in \{1, 2, \ldots, k\}$ **do**

3: $\quad \mathbf{f}_j \leftarrow$LPME$\Big(\mathcal{S}_{\mathbf{z}_j}, \epsilon, \Omega\Big)$ with $\mathcal{S}_{\mathbf{z}_j} \subset \mathcal{S}$ and obtain (7)

4: $\mathbf{f}_1^- \leftarrow$ LPME$(\mathcal{S}_{-\mathbf{z}_1}, \epsilon, \Omega)$ with $\mathcal{S}_{-\mathbf{z}_1} \subset \mathcal{S}$ and obtain (8)

5: $\hat{\mathbf{a}}, \hat{\mathbf{B}} \leftarrow$ normalized solution dervied from (9)

Output: $\hat{\mathbf{a}}, \hat{\mathbf{B}}$

4 ELICITING FAIRNESS METRICS

Having understood the QPME procedure, we now discuss our main application of *quadratic metric elicitation for algorithmic fairness*. Like Hiranandani et al. [2020], we consider eliciting a metric that trades-off between predictive performance and fairness violation [Kamishima et al., 2012, Chouldechova, 2017, Menon and Williamson, 2018]. However, unlike Hiranandani et al. [2020], we handle general quadratic fairness violations and show how QPME can be easily employed to elicit group-fair metrics.

4.1 FAIRNESS PRELIMINARIES

The fairness setting is the same as the one in Section 3 except that we additionally have m groups in the data and use $g \in [m]$ to denote the group membership. The groups are assumed to be disjoint, fixed, and known apriori [Agarwal et al., 2018]. We will work with a separate (randomized) classifiers $h^g : \mathcal{X} \to \Delta_k$ for each group g, and use $\mathcal{H}^g = \{h^g : \mathcal{X} \to \Delta_k\}$ to denote the set of all classifiers for g.

Group predictive rates: Similar to (1), we denote the group-conditional rates for h^g by $\mathbf{r}^g(h^g, \mathbb{P}) \in \mathbb{R}^k$, where the i-th entry is additionally conditioned on group g:

$$r_i^g(h^g, \mathbb{P}) := \mathbb{P}(h^g = i | Y = i, G = g)\, \forall\, i \in [k]. \tag{10}$$

Analogous to the general setup, we denote the set of feasible rates for group g by $\mathcal{R}^g = \{\mathbf{r}^g(h^g, \mathbb{P}) : h^g \in \mathcal{H}^g\}$.

Rates for overall classifier: We construct the overall classifier $h : (\mathcal{X}, [m]) \to \Delta_k$ by predicting with classifier h^g for group g, i.e. $h(\mathbf{x}, g) := h^g(\mathbf{x})$. We will be interested in both the fairness violation and predictive performance of the overall classifier. For the former, we will need the m group-specific rates, represented together as a tuple:

$$\mathbf{r}^{1:m} := (\mathbf{r}^1, \ldots, \mathbf{r}^m) \in \mathcal{R}^1 \times \cdots \times \mathcal{R}^m =: \mathcal{R}^{1:m}.$$

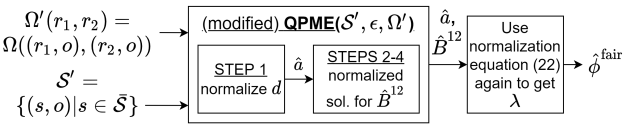


Figure 3: Eliciting Fair Quadratic Metrics for two groups. We formulate a k-dimensional elicitation problem and use a variant of QPME (Algorithm 1).

For the latter, we will measure the overall rates for h as described in (1). The overall rates can also be written in terms of group-specific rates as: $\mathbf{r} = \sum_{g=1}^{m} \boldsymbol{\tau}^g \odot \mathbf{r}^g$, where $\boldsymbol{\tau}^g$ is just a constant vector whose i-th entry denote the prevalence of group g within class i, i.e., $\mathbb{P}(G = g | Y = i)$.

4.2 FAIR QUADRATIC METRIC ELICITATION

We seek to elicit a metric that trades-off between predictive performance (a linear function of overall rates $\mathbf{r}$) and fairness violation (a quadratic function of group rates $\mathbf{r}^{1:m}$). For simplicity, we will denote the fairness metric in cost form, i.e., lower values are better.

Definition 4. (Fair Quadratic Performance Metric) *For misclassification costs $\mathbf{a} \in \mathbb{R}^k$, $\mathbf{a} \geq 0$, fairness violation costs $\mathbb{B} = \{\mathbf{B}^{uv} \in PSD_k\}_{u,v=1,v>u}^{m}$, and a trade-off parameter $\lambda \in [0, 1]$, we define:*

$$\phi^{\text{fair}}(\mathbf{r}^{1:m}; \mathbf{a}, \mathbb{B}, \lambda) := (1-\lambda)\langle \mathbf{a}, \mathbf{1} - \mathbf{r} \rangle + \lambda \frac{1}{2} \left(\sum\nolimits_{v>u} (\mathbf{r}^u - \mathbf{r}^v)^T \mathbf{B}^{uv} (\mathbf{r}^u - \mathbf{r}^v) \right), \quad (11)$$

where w.l.o.g. the parameters $\mathbf{a}$ and $\mathbf{B}^{uv}$'s are normalized: $\|\mathbf{a}\|_2 = 1$, $\frac{1}{2}\sum_{v>u}^{m} \|\mathbf{B}^{uv}\|_F = 1$.

The coefficients $\mathbf{a}, \mathbf{B}^{uv}$'s are separately normalized so that the predictive performance and fairness violation are in the same scale, and we can additionally elicit the trade-off parameter λ. Analogous to Definitions 1–2, the problem of *Fair Quadratic Metric Elicitation* is as follows: given access to pairwise oracle queries of the form $\Omega(\hat{\mathbf{r}}_1^{1:m}, \hat{\mathbf{r}}_2^{1:m})$, recover a metric $\hat{\phi}^{\text{fair}} = (\hat{\mathbf{a}}, \hat{\mathbb{B}}, \hat{\lambda})$ such that $\|\phi^{\text{fair}} - \hat{\phi}^{\text{fair}}\| < \kappa$ under a suitable norm $\|\cdot\|$ for small $\kappa > 0$.

Similar to Section 2.2, we study the space of feasible rates $\mathcal{R}^{1:m}$ under the following mild assumption.

Assumption 3. *For all $g \in [m]$, the conditional distributions $P(Y = j | X, G = g)$, $j \in [k]$, are distinct, i.e., there is some signal for non-trivial classification for each group.*

Proposition 2 (Geometry of $\mathcal{R}^{1:m}$; Figure 2(b))**.** *For each group g, a classifier that predicts class i on all inputs results in the same rate vector $\mathbf{e}_i$. The rate space $\mathcal{R}^g$ for each group g is convex and so is the intersection $\mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$, which also contains the rate profile $\mathbf{o} = \frac{1}{k}\sum_{i=1}^{k} \mathbf{e}_i$ (achieved by the uniform random classifier) in the interior.*

Remark 3 (Existence of sphere $\bar{\mathcal{S}}$)**.** *There exists a sphere $\bar{\mathcal{S}} \subset \mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$ of radius ρ centered at $\mathbf{o}$. Thus, a rate $\mathbf{s} \in \bar{\mathcal{S}}$ is feasible for each of the m groups, i.e., $\mathbf{s}$ is achievable by some classifier h^g for each group $g \in [m]$.*

Because we allow a separate classifier for each group, Remark 3 implies that any rate $\mathbf{r}^{1:m} = (\mathbf{s}^1, \ldots, \mathbf{s}^m)$ for arbitrary points $\mathbf{s}^1, \ldots, \mathbf{s}^m \in \bar{\mathcal{S}}$ is achievable for some choice of group-specific classifiers $h^1, \ldots, h^m$. This observation will be key to the elicitation algorithm we describe next.

4.3 ELICITING METRIC PARAMETERS $(\mathbf{a}, \mathbb{B}, \lambda)$

We present a strategy for eliciting fair metrics by adapting the QPME algorithm. For simplicity, we focus on the $m = 2$ case and extend our approach for $m > 2$ in Appendix D.

Observe that for a rate profile $\mathbf{r}^{1:2} = (\mathbf{s}, \mathbf{o})$, where the first group is assigned an arbitrary point in $\bar{\mathcal{S}}$ and the second group is assigned the uniform random classifier's rate $\mathbf{o}$, the fair metric (11) becomes: $\phi^{\text{fair}}((\mathbf{s}, \mathbf{o}); \mathbf{a}, \mathbf{B}^{12}, \lambda)$

$$\begin{aligned} &:= (1-\lambda)\langle \mathbf{a}, \mathbf{1} - (\boldsymbol{\tau}^1 \odot \mathbf{s} + \boldsymbol{\tau}^2 \odot \mathbf{o}) \rangle + \frac{\lambda}{2}(\mathbf{s} - \mathbf{o})^T \mathbf{B}^{12} (\mathbf{s} - \mathbf{o}) \\ &:= \langle \mathbf{d}, \mathbf{s} - \mathbf{o} \rangle + \frac{1}{2}(\mathbf{s} - \mathbf{o})^T \mathbf{B} (\mathbf{s} - \mathbf{o}) \\ &:= \bar{\phi}(\mathbf{s}; \mathbf{d}, \mathbf{B}), \end{aligned} \quad (12)$$

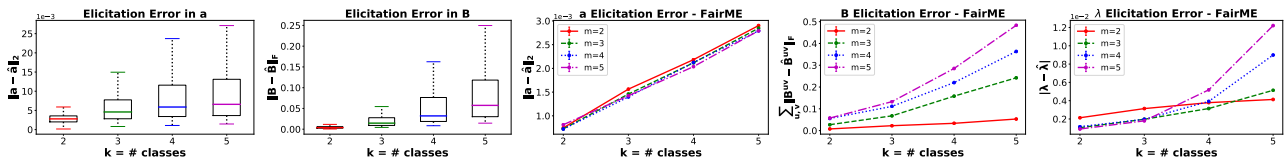
where $\mathbf{d} = -(1-\lambda)\boldsymbol{\tau}^1 \odot \mathbf{a}$ and $\mathbf{B} = \lambda \mathbf{B}^{12}$, and we use $\boldsymbol{\tau}^1 + \boldsymbol{\tau}^2 = \mathbf{1}$ (the vector of ones) for the second step. The metric $\bar{\phi}$ above is a particular instance of the quadratic metric in (4). We can thus apply a slight variant of the QPME procedure in Algorithm 1 to solve the quadratic metric elicitation problem over the sphere $\mathcal{S}' = \{(\mathbf{s}, \mathbf{o}) \,|\, \mathbf{s} \in \bar{\mathcal{S}}\}$ with the modified oracle $\Omega'(\mathbf{r}_1, \mathbf{r}_2) = \Omega((\mathbf{r}_1, \mathbf{o}), (\mathbf{r}_2, \mathbf{o}))$.

The only change needed for the algorithm is in line 5, where we need to account for the changed relationship between $\mathbf{d}$ and $\mathbf{a}$ and need to separately (not jointly) normalize the linear and quadratic coefficients. With this change, the output of the algorithm directly gives us the required estimates. Specifically, from step 1 of Algorithm 1 and (6), we have $\hat{d}_i = -(1-\lambda)\tau_i^1 \hat{a}_i$. By normalizing $\mathbf{d}$, we get $\hat{\mathbf{a}} = \frac{\mathbf{d}}{\|\mathbf{d}\|}$ for the linear coefficients. Similarly, steps 2-4 of Algorithm 1 and (9) allow us to express $\hat{B}_{ij} = \lambda \hat{B}_{ij}^{12}$ in terms of $\hat{a}_1$. After normalizing we directly get estimates $\hat{\mathbf{B}}^{12} = \hat{\mathbf{B}}/\|\hat{\mathbf{B}}\|_F$ for the quadratic coefficients.

Finally, because the linear and quadratic coefficients are separately normalized, the estimates $\hat{\mathbf{a}}$, $\hat{\mathbf{B}}^{12}$ are independent of the trade-off parameter λ. Given estimates $\hat{B}_{ij}^{12}$ and $\hat{a}_1$, we can now additionally estimate the trade-off parameter $\hat{\lambda}$. See Appendix D for details and Figure 3 for an illustration.

5 GUARANTEES

We discuss guarantees for the QPME procedure under the following practically relevant feedback model. The fair metric elicitation guarantees follow as a consequence.



(a) (b) (c) (d) (e)

Figure 4: Average elicitation error over 100 metrics as a function of number of classes k and groups m for quadratic metrics in Definition 3 (a–b) and fairness metrics in Definition 4 (c–e). See Table 1 in Appendix I for the number of queries needed.

Definition 5 (Oracle Feedback Noise: $\epsilon_\Omega \geq 0$)**.** *Given rates* $\mathbf{r}_1, \mathbf{r}_2$*, the oracle responds correctly iff* $|\phi^{\text{quad}}(\mathbf{r}_1) - \phi^{\text{quad}}(\mathbf{r}_2)| > \epsilon_\Omega$ *and may be incorrect otherwise.*

In words, the oracle may respond incorrectly if the rates are close as measured by the metric. Since eliciting the metric involves offline computations of ratios, we make a regularity assumption ensuring that all components are well defined.

Assumption 4. *For the shifted quadratic metric* $\bar{\phi}$ *in* (4)*, the gradients at the rate profiles* $\mathbf{o}$*,* $-\mathbf{z}_1$*, and* $\{\mathbf{z}_1, \ldots, \mathbf{z}_q\}$*, are non-zero vectors. Additionally,* $\rho > \varrho \gg \epsilon_\Omega$*.*

Theorem 1. *Given* $\epsilon, \epsilon_\Omega \geq 0$*, and a 1-Lipschitz metric* ϕ^{quad} *(Def. 3) parametrized by* $\mathbf{a}, \mathbf{B}$*, under Assumptions 1, 2, and 4, after* $O\left(k^2 \log \frac{1}{\epsilon}\right)$ *queries, Algorithm 1 returns a metric* $\hat{\phi}^{\text{quad}} = (\hat{\mathbf{a}}, \hat{\mathbf{B}})$ *with* $\|\mathbf{a} - \hat{\mathbf{a}}\|_2 \leq O\left(\sqrt{k}(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho})\right)$ *and* $\|\mathbf{B} - \hat{\mathbf{B}}\|_F \leq O\left(k\sqrt{k}(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho})\right)$*.*

The proof of Theorem 1 uses the guarantee for LPME *only* as an intermediate step, and substantially builds on it to take into account the smoothness of the non-linear metric, the multiplicative errors in the slopes, and the feedback noise. We also provide a *finite sample version* of Theorem 1 in Corollary 1 (Appendix G), which states that the above result holds with high probability as long as (i) the hypothesis class of classifiers has finite capacity, and (ii) the number of samples used to estimate the rates is large enough.

Theorem 2. ***(Lower Bound)*** *For any* $\epsilon > 0$*, at least* $\Omega(k^2 \log(1/k\sqrt{k}\epsilon))$ *pairwise queries are needed to to elicit a quadratic metric (Def. 3) to an error tolerance of* $k\sqrt{k}\epsilon$*.*

Theorem 1 shows that the QPME procedure is robust to noise and its query complexity depends only *linearly* in the number of unknowns. Theorem 2 shows that the inherent complexity of the problem depends on the *number of unknowns*, thus our query complexity is optimal (barring the log term). So the $\tilde{O}(k^2)$ complexity is merely an artifact of our setup in Definition 3 being very general (with $O(k^2)$ unknowns). Indeed, with added structural assumptions on the metric, our proposal can be modified to considerably reduce the query complexity. For example, if we know that the matrix $\mathbf{B}$ is diagonal, then each LPME subroutine call needs to estimate only one parameter, which can be done with a constant number of queries, requiring a total of only $\tilde{O}(k)$ queries. We also stress that despite eliciting a more complex (non-linear) metric, the query complexity is still *linear in the number of unknowns*, which is same as prior linear elicitation methods [Hiranandani et al., 2019a,b].

6 EXPERIMENTS

We evaluate our approach on simulated oracles. Here we present results on a synthetically generated query space and in Appendix I.2 include results on real-world datasets.

Eliciting quadratic metrics. We first apply QPME (Algorithm 1) to elicit quadratic metrics in Definition 3. Like Hiranandani et al. [2020], we assume access to a k-dimensional sphere $\mathcal{S}$ centered at rate $\mathbf{o}$ with radius $\rho = 0.2$, from which we query rate vectors $\mathbf{r}$. The trends that we will discuss are robust to the sphere radius parameter ρ. Recall that in practice, Remark 1 guarantees the existence of such a sphere within the feasible region $\mathcal{R}$. We randomly generate quadratic metrics ϕ^{quad} parametrized by $(\mathbf{a}, \mathbf{B})$ and repeat the experiment over 100 trials for varying numbers of classes $k \in \{2, 3, 4, 5\}$. We run the QPME procedure with tolerance $\epsilon = 10^{-2}$. In Figures 4(a)–4(b), we show box plots of the ℓ_2 (Frobenius) norm between the true and elicited linear (quadratic) coefficients. We generally find that QPME is able to elicit metrics close to the true ones. This holds for varying k, showing the effectiveness of our approach in handling multiple classes. The average number of queries we needed for elicitation over the 100 trials is provided in Table 1 in Appendix I. Note that the number of queries is $\tilde{O}(d)$ for eliciting a quadratic metric with $d = k^2$ unknowns, which clearly matches the lower bound in Theorem 2. See Appendix F for a discussion on the practicality of posing the requisite number of queries.

Eliciting fairness metrics. We next apply the elicitation procedure in Figure 3 with tolerance $\epsilon = 10^{-2}$ to elicit the fairness metrics in Definition 4. We randomly generate oracle metrics ϕ^{fair} parametrized by $(\mathbf{a}, \mathbb{B}, \lambda)$ and repeat the experiment over 100 trials and with varied number of classes and groups $k, m \in \{2, 3, 4, 5\}$. Figures 4(c)–4(e) show the mean elicitation errors for the the three parameters. For the linear predictive performance, the error $\|\mathbf{a} - \hat{\mathbf{a}}\|_2$ increases only with the number of classes k and not groups m, as it is independent of the number of groups. For the quadratic violation term, the error $\sum_{u,v} \|\mathbf{B}^{uv} - \hat{\mathbf{B}}^{uv}\|_F$ increases with both k and m. This is because the QPME procedure is run $\binom{m}{2}$ times for eliciting $\binom{m}{2}$ matrices $\{\mathbf{B}^{uv}\}_{v>u}$, and so the elicitation error accumulates with increasing q. Lastly, the elicited trade-off $\hat{\lambda}$ is seen to be close to the true λ as well.

Real-world datasets. In App. I.2, we evaluate how well the elicited metric from QPME ranks a set of candidate classifiers trained on real-world datasets. We find that despite incurring elicitation errors, QPME achieves near-perfect ranking; whereas, the baseline metrics fail to do so.

7 RELATED WORK

Hiranandani et al.[2019a] formalized the problem of ME for binary classification with (quasi-)linear metrics and later extended it to the multiclass setting [Hiranandani et al., 2019b]. Unlike them, we elicit more complex quadratic metrics, and also provide an information-theoretic lower bound on the query complexity (Theorem 2). Prior works on ME offer no such lower bound guarantees. Learning linear functions passively using pairwise comparisons is a mature field [Joachims, 2002, Peyrard et al., 2017], but unlike their active learning counter-parts [Settles, 2009, Kane et al., 2017], these methods are not query-efficient. Studies such as Qian et al. [2015] provide active linear elicitation strategies but with no guarantees and also work with a different query space. We are unaware of prior work that *provably* elicit a quadratic function, either passively or actively using pairwise comparisons. Our work is thus a significant first step towards active, nonlinear metric elicitation.

The use of metric elicitation for fairness is relatively new, with some work on eliciting *individual* fairness metrics [Ilvento, 2020, Mukherjee et al., 2020]. Hiranandani et al. [2020] is the only work we are aware of that elicits *group-fair* metrics, which we extend to handle more general metrics. Zhang et al. [2020] elicit the trade-off between accuracy and fairness using complex ratio queries. In contrast, we jointly elicit the predictive performance, fairness violation, and trade-off using simpler pairwise queries. Lastly, there has been work on learning fair classifiers under constraints [Zafar et al., 2017, Agarwal et al., 2018]. We take the regularization view of fairness, where the fairness violation is included in the objective [Kamishima et al., 2012].

Our work is also related to *preference elicitation* in the decision-theory literature [Boutilier et al., 2006, Benabbou et al., 2017, Perny et al., 2016, Braziunas, 2012, Benabbou et al., 2017], *learning reward functions* in the inverse reinforcement learning literature [Wu et al., 2020, Abbeel and Ng, 2004, Levine et al., 2011, Sadigh et al., 2017], and the *Bradley-Terry-Luce model with features* in the learning-to-rank literature [Shah et al., 2015, Niranjan and Rajkumar, 2017]. In summary, these studies focus on either eliciting linear utilities or passively learning utility functions. Our work is substantially different from them as we are tied to the geometry of the space of classification error statistics, and elicit quadratic utility functions using only pairwise comparisons, and particularly, in an active learning fashion. Moreover, we also provide query complexity bounds along with a lower bound. We further elaborate on the specific differences from these papers in Appendix J.

8 DISCUSSION

We have provided an efficient quadratic metric elicitation strategy with application to fairness, and with a query complexity that has the same dependence on the number of unknowns as that for linear metrics. We show in Appendix H how our approach can be extended to elicit *higher-order polynomial* metrics. Thus our work not only increases the use-cases for ME but also opens the door for non-linear metric elicitation in other fields such as *active learning*.

Handling large number of classes: For applications where k is very large, the parameterization discussed in Section 2 may not be applicable in its current form. For example, when $k = 1000$, the quadratic metric in (3) would use $O(1000^2)$ parameters, an exorbitantly high number to elicit in practice. Note that the presence of k^2 unknowns is an artifact of the problem formulation, and *not* of our proposed procedure. Moreover, as shown in Theorem 2, it is *theoretically impossible* to estimate k^2 unknowns with fewer than $\tilde{O}(k^2)$ queries. While our QPME procedure does indeed match this lower bound, in practice, we do not expect it to be applied to estimate such an over-parameterized metric. Instead, for such large-scale settings, we recommend making reasonable assumptions on the metric to reduce the number of unknowns, e.g., by having multiple classes share the same parameter, and the query complexity of QPME would then only depend *linearly* on the reduced number of unknowns. For instance, in Table 4 (Appendix I.2), we show that by simplifying the metric with structural assumptions, one can use fewer queries in practice to get comparable results.

Advantages: Our proposal comes with many practical advantages: (a) *Fairness:* we are aware of no prior work that can elicit fair quadratic metrics, particularly with provable guarantees; (b) *Transportability:* our method is independent of the population $\mathbb{P}$, which allows any metric that is elicited using one dataset or model class to be applied to other applications, as long as the expert believes the tradeoffs to be the same; and (c) *Feasibility:* we ensure that the rates are feasible throughout the elicitation (i.e., are achievable by classifiers), which allows the flexibility to deploy systems that either compare classifiers or compare rates.

Limitations: Limitations of our work include the assumption that the metric has a parametric form, which can be restrictive in some cases, and not providing a concrete answer to who the oracles should be. One should also be cautious in applying ME to eliciting fairness metrics, as failures here could exacerbate the adverse effects on protected groups.

Future Work: To thoroughly answer the above questions, we are actively conducting user studies on collecting preference feedback using intuitive visualizations of rates [Shen et al., 2020, Beauxis-Aussalet and Hardman, 2014] or classifiers [Ribeiro et al., 2016] . Please see Appendix K to take a peek into the future work, where we discuss findings from a preliminary user study.

References

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *ICML*, 2004.

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudik, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *ICML*, 2018.

Emma Beauxis-Aussalet and Lynda Hardman. Visualization of confusion matrix for non-expert users. In *IEEE Conference on Visual Analytics Science and Technology, Poster Proceedings*, 2014.

Yahav Bechavod and Katrina Ligett. Learning fair classifiers: A regularization-inspired approach. In *4th Workshop on FAccT in Machine Learning*, 2017.

Nawal Benabbou, Patrice Perny, and Paolo Viappiani. Incremental elicitation of choquet capacities for multicriteria choice, ranking and sorting problems. *Artificial Intelligence*, 246:152–180, 2017.

Craig Boutilier, Relu Patrascu, Pascal Poupart, and Dale Schuurmans. Constraint-based optimization and utility elicitation using the minimax decision criterion. *Artificial Intelligence*, 170(8-9):686–713, 2006.

Darius Braziunas. *Decision-theoretic elicitation of generalized additive utilities*. PhD thesis, 2012.

Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163, 2017.

Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the ERM principle. *JMLR*, 16(1):2377–2404, January 2015.

Pavel Dmitriev and Xian Wu. Measuring metrics. In *CIKM*, 2016.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *ITCS*, pages 214–226, 2012.

A. Esuli and F. Sebastiani. Optimizing text quantifiers for multivariate loss functions. *ACM Transactions on Knowledge Discovery and Data*, 9(4), 2015.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *ArXiv preprint, arXiv:1710.11248*, 2017.

Johannes Fürnkranz and Eyke Hüllermeier. Preference learning and ranking by pairwise comparison. In *Preference Learning*, pages 65–82. Springer, 2010.

Gabriel Goh, Andrew Cotter, Maya Gupta, and Michael P Friedlander. Satisfying real-world goals with dataset constraints. In *NeurIPS*, 2016.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *NeurIPS*, pages 3315–3323, 2016.

Gaurush Hiranandani, Shant Boodaghians, Ruta Mehta, and Oluwasanmi Koyejo. Performance metric elicitation from pairwise classifier comparisons. In *AISTATS*, 2019a.

Gaurush Hiranandani, Shant Boodaghians, Ruta Mehta, and Oluwasanmi O Koyejo. Multiclass performance metric elicitation. In *NeurIPS*, 2019b.

Gaurush Hiranandani, Harikrishna Narasimhan, and Oluwasanmi Koyejo. Fair performance metric elicitation. In *NeurIPS*, 2020.

Christina Ilvento. Metric learning for individual fairness. 2020.

Thorsten Joachims. Svm-light: Support vector machine. *URL: `http://svmlight.joachims.org/`, University of Dortmund*, 1999.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *ACM SIGKDD*, 2002.

Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *ECML-PKDD*, pages 35–50. Springer, 2012.

Daniel M Kane, Shachar Lovett, Shay Moran, and Jiapeng Zhang. Active classification with comparison queries. In *FOCS*, 2017.

Purushottam Kar, Shuai Li, Harikrishna Narasimhan, Sanjay Chawla, and Fabrizio Sebastiani. Online optimization methods for the quantification problem. In *ACM SIGKDD*, 2016.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *NeurIPS*, 2017.

David G Kleinbaum, K Dietz, M Gail, Mitchel Klein, and Mitchell Klein. *Logistic regression*. Springer, 2002.

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *ITCS*, 2017.

Oluwasanmi O Koyejo, Nagarajan Natarajan, Pradeep K Ravikumar, and Inderjit S Dhillon. Consistent multilabel classification. In *NeurIPS*, 2015.

Brian Kulis et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013.

Sergey Levine, Zoran Popovic, and Vladlen Koltun. Nonlinear inverse reinforcement learning with gaussian processes. *NeurIPS*, 24:19–27, 2011.

Bruce G Lindsay, Marianthi Markatou, Surajit Ray, Ke Yang, Shu-Chuan Chen, et al. Quadratic distances on probabilities: A unified foundation. *The Annals of Statistics*, 36 (2):983–1006, 2008.

Manuel Lopes, Francisco Melo, and Luis Montesano. Active learning for reward estimation in inverse reinforcement learning. In *ECML-PKDD*, pages 31–46. Springer, 2009.

Aditya Menon, Harikrishna Narasimhan, Shivani Agarwal, and Sanjay Chawla. On the statistical consistency of algorithms for binary classification under class imbalance. In *ICML*, 2013.

Aditya Krishna Menon and Robert C Williamson. The cost of fairness in binary classification. In *FAccT*, 2018.

Soheil Mohajer, Changho Suh, and Adel Elmahdy. Active learning for top-k rank aggregation from noisy comparisons. In *ICML*, 2017.

Debarghya Mukherjee, Mikhail Yurochkin, Moulinath Banerjee, and Yuekai Sun. Two simple ways to learn individual fairness metric from data. In *ICML*, 2020.

Harikrishna Narasimhan. Learning with complex loss functions and constraints. In *AISTATS*, 2018.

Harikrishna Narasimhan, Harish Ramaswamy, Aadirupa Saha, and Shivani Agarwal. Consistent multiclass algorithms for complex performance measures. In *ICML*, 2015.

B. K. Natarajan. On learning sets and functions. *Machine Learning*, 4(1):67–97, 1989.

Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *ICML*, 2000.

UN Niranjan and Arun Rajkumar. Inductive pairwise ranking: going beyond the n log (n) barrier. In *AAAI*, volume 31, 2017.

SK Pal and S Mitra. Multilayer perceptron, fuzzy sets, and classification. *IEEE Transactions on Neural Networks*, 3 (5):683–697, 1992.

Patrice Perny, P. Viappiani, and A. Boukhatem. Incremental preference elicitation for decision making under risk with the rank-dependent utility model. In *UAI*, 2016.

Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. Learning to score system summaries for better content selection evaluation. In *Proceedings of the Workshop on New Frontiers in Summarization*, 2017.

Buyue Qian, Xiang Wang, Fei Wang, Hongfei Li, Jieping Ye, and Ian Davidson. Active learning from relative queries. In *IJCAI*, 2013.

Li Qian, Jinyang Gao, and HV Jagadish. Learning user preferences by adaptive pairwise comparison. *Proceedings of the VLDB Endowment*, 8(11):1322–1333, 2015.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *ACM SIGKDD*, 2016.

Dorsa Sadigh, Anca D Dragan, Shankar Sastry, and Sanjit A Seshia. Active preference-based learning of reward functions. In *RSS*, 2017.

Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

Nihar Shah, S. Balakrishnan, J. Bradley, A. Parekh, K. Ramchandran, and Martin Wainwright. Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. In *AISTATS*, 2015.

Hong Shen, H. Jin, A. A. Cabrera, A. Perer, H. Zhu, and Jason I Hong. Designing alternative representations of confusion matrices to support non-expert public understanding of algorithm performance. *CSCW2*, 2020.

Grace S Shieh. A weighted kendall's tau statistic. *Statistics & probability letters*, 39(1):17–24, 1998.

Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

Marshall H Stone. The generalized weierstrass approximation theorem. *Mathematics Magazine*, 21(5):237–254, 1948.

Giordano Tamburrelli and Alessandro Margara. Towards automated *A/B* testing. In *International Symposium on Search Based Software Engineering*, 2014.

H. Valizadegan, R. Jin, R. Zhang, and J. Mao. Learning to rank by optimizing NDCG measure. In *NeurIPS*, 2009.

Zheng Wu, Liting Sun, Wei Zhan, Chenyu Yang, and Masayoshi Tomizuka. Efficient sampling-based maximum entropy inverse reinforcement learning with application to autonomous driving. *IEEE Robotics and Automation Letters*, 5(4):5355–5362, 2020.

Sitan Yang and Daniel Q Naiman. Multiclass cancer classification based on gene expression comparison. *Statistical Applications in Genetics and Molecular Biology*, 13(4): 477–496, 2014.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *AISTATS*, 2017.

Yunfeng Zhang, Rachel Bellamy, and Kush Varshney. Joint optimization of ai fairness and utility: A human-centered approach. In *AIES*, pages 400–406, 2020.

Appendix for "Quadratic Metric Elicitation for Fairness and Beyond"

A LINEAR PERFORMANCE METRIC ELICITATION (LPME)

In this section, we shed more light on the procedure from [Hiranandani et al., 2019b] that elicits a multiclass linear metric. We call it the Linear Performance Metric Elicitation (LPME) procedure. As discussed in Algorithm 1, we use this as a subroutine to elicit metrics in the quadratic family.

LPME exploits the enclosed sphere $\mathcal{S} \subset \mathcal{R}$ for eliciting linear multiclass metrics. Let the sphere $\mathcal{S}$'s radius be $\rho > 0$, and the oracle's scale invariant metric be $\phi^{\text{lin}}(\mathbf{r}) := \langle \mathbf{a}, \mathbf{r} \rangle$ such that $\|\mathbf{a}\|_2 = 1$. The oracle queries are $\Omega\left(\mathbf{r}_1, \mathbf{r}_2\,;\, \phi^{\text{lin}}\right) := \mathbb{1}[\phi^{\text{lin}}(\mathbf{r}_1) > \phi^{\text{lin}}(\mathbf{r}_2)]$. We first outline a trivial Lemma from [Hiranandani et al., 2019b].

Lemma 1. *[Hiranandani et al., 2019b] Let a normalized vector $\mathbf{a}$ with $\|\mathbf{a}\|_2 = 1$ parametrize a linear metric $\phi^{lin} := \langle \mathbf{a}, \mathbf{r} \rangle$, then the unique optimal rate $\bar{\mathbf{r}}$ over $\mathcal{S}$ is a rate on the boundary of $\mathcal{S}$ given by $\bar{\mathbf{r}} = \rho\mathbf{a} + \mathbf{o}$, where $\mathbf{o}$ is the center of $\mathcal{S}$.*

Algorithm 2 Linear Performance Metric Elicitation

Input: Query space $\mathcal{S} \subset \mathcal{R}$, binary-search tolerance $\epsilon > 0$, oracle $\Omega(\cdot, \cdot\,;\, \phi^{\text{lin}})$ with metric ϕ^{lin}

for $i = 1, 2, \cdots k$ **do**
  Set $\mathbf{a} = \mathbf{a}' = (1/\sqrt{k}, \ldots, 1/\sqrt{k})$.
  Set $a'_i = -1/\sqrt{k}$.
  Compute the optimal $\bar{s}^{(\mathbf{a})}$ and $\bar{s}^{(\mathbf{a}')}$ over the sphere $\mathcal{S}$ using Lemma 1
  Query $\Omega(\bar{\mathbf{s}}^{(\mathbf{a})}, \bar{\mathbf{s}}^{(\mathbf{a}')}; \phi^{\text{lin}})$
  {These queries reveal the search orthant}

Start with coordinate $j = 1$.
Initialize: $\boldsymbol{\theta} = \boldsymbol{\theta}^{(1)}$ {$\boldsymbol{\theta}^{(1)}$ is a point in the search orthant.}
for $t = 1, 2, \cdots, T = 3(k-1)$ **do**
  Set $\boldsymbol{\theta}^{(a)} = \boldsymbol{\theta}^{(c)} = \boldsymbol{\theta}^{(d)} = \boldsymbol{\theta}^{(e)} = \boldsymbol{\theta}^{(b)} = \boldsymbol{\theta}^{(t)}$.
  Set $\theta_j^{(a)}$ and $\theta_j^{(b)}$ to be the min and max angle, respectively, based on the search orthant
  while $\left|\theta_j^{(b)} - \theta_j^{(a)}\right| > \epsilon$ **do**
    Set $\theta_j^{(c)} = \frac{3\theta_j^{(a)} + \theta_j^{(b)}}{4}$, $\theta_j^{(d)} = \frac{\theta_j^{(a)} + \theta_j^{(b)}}{2}$, and $\theta_j^{(e)} = \frac{\theta_j^{(a)} + 3\theta_j^{(b)}}{4}$.
    Set $\bar{\mathbf{r}}^{(a)} = \mu(\boldsymbol{\theta}^{(a)})$ (i.e. parametrization of $\partial\mathcal{S}$). Similarly, set $\bar{\mathbf{r}}^{(c)}, \bar{\mathbf{r}}^{(d)}, \bar{\mathbf{r}}^{(e)}, \bar{\mathbf{r}}^{(b)}$
    $[\theta_j^{(a)}, \theta_j^{(b)}] \leftarrow$ *ShrinkInterval* $(\Omega, \bar{\mathbf{r}}^{(a)}, \bar{\mathbf{r}}^{(c)}, \bar{\mathbf{r}}^{(d)}, \bar{\mathbf{r}}^{(e)}, \bar{\mathbf{r}}^{(b)})$ {see Figure 5}
  Set $\theta_j^{(d)} = \frac{1}{2}(\theta_j^{(a)} + \theta_j^{(b)})$
  Set $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(d)}$.
  Update coordinate $j \leftarrow j + 1$ cyclically.
Output: $\hat{a}_i = \Pi_{j=1}^{i-1} \sin\theta_j^{(T)} \cos\theta_i{}^{(T)}\, \forall i \in [k-1],\ \hat{a}_k = \Pi_{j=1}^{k-1} \sin\theta_j^{(T)}$

Lemma 1 provides a way to define a one-to-one correspondence between a linear performance metric and its optimal rate over the sphere. That is, given a linear performance metric, using Lemma 1, we may get a unique point in the query space lying on the boundary of the sphere $\partial\mathcal{S}$. Moreover, the converse is also true; i.e., given a feasible rate on the boundary of the sphere $\partial\mathcal{S}$, one may recover the linear metric for which the given rate is optimal. Thus, for eliciting a linear metric, Hiranandani et al. [2019b] essentially search for the optimal rate (over the sphere $\mathcal{S}$) using pairwise queries to the oracle. The optimal rate by virtue of Lemma 1 reveals the true metric. The LPME subroutine is summarized in Algorithm 2. Intuitively, Algorithm 2 minimizes a strongly convex function denoting distance of query points from a supporting hyperplane whose slope is the true metric (see Figure 2(c) in [Hiranandani et al., 2019b]). The procedure also uses the following standard paramterization for the surface of the sphere $\partial\mathcal{S}$.

Parameterizing the boundary of the enclosed sphere $\partial\mathcal{S}$. Let $\boldsymbol{\theta}$ be a $(k-1)$-dimensional vector of angles. In $\boldsymbol{\theta}$, all the angles except the primary angle are in $[0, \pi]$, and the primary angle is in $[0, 2\pi]$. A scale invariant linear performance metric

Subroutine *ShrinkInterval*
Input: Oracle Ω and rate profiles $\bar{\mathbf{r}}^{(a)}, \bar{\mathbf{r}}^{(c)}, \bar{\mathbf{r}}^{(d)}, \bar{\mathbf{r}}^{(e)}, \bar{\mathbf{r}}^{(b)}$
Query $\Omega(\bar{\mathbf{r}}^{(a)}, \bar{\mathbf{r}}^{(c)}\,;\,\phi^{\text{lin}})$.
If $(\bar{\mathbf{r}}^{(a)} \succ \bar{\mathbf{r}}^{(c)})$ Set $\theta_j^{(b)} = \theta_j^{(d)}$.
else Query $\Omega(\bar{\mathbf{r}}^{(c)}, \bar{\mathbf{r}}^{(d)}\,;\,\phi^{\text{lin}})$.
  If $(\bar{\mathbf{r}}^{(c)} \succ \bar{\mathbf{r}}^{(d)})$ Set $\theta_j^{(b)} = \theta_j^{(d)}$.
  else Query $\Omega(\bar{\mathbf{r}}^{(d)}, \bar{\mathbf{r}}^{(e)}\,;\,\phi^{\text{lin}})$.
    If $(\bar{\mathbf{r}}^{(d)} \succ \bar{\mathbf{r}}^{(e)})$ Set $\theta_j^{(a)} = \theta_j^{(c)}$ and $\theta_j^{(b)} = \theta_j^{(e)}$.
    else Set $\theta_j^{(a)} = \theta_j^{(c)}$.
Output: $[\theta_j^{(a)}, \theta_j^{(b)}]$.

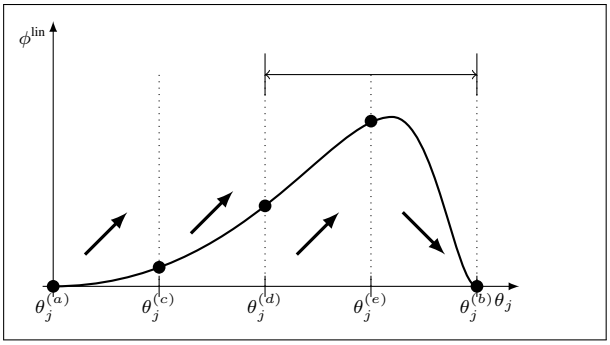


Figure 5: (Left): The *ShrinkInterval* subroutine used in line 16 of Algorithm 2 (Right): Visual illustration of the subroutine *ShrinkInterval* [Hiranandani et al., 2019b]; *ShrinkInterval* shrinks the current search interval to half based on oracle responses to at most three queries.

with $\|\mathbf{a}\|_2 = 1$ can be constructed by assigning $a_i = \Pi_{j=1}^{i-1} \sin\theta_j \cos\theta_i$ for $i \in [k-1]$ and $a_k = \Pi_{j=1}^{k-1} \sin\theta_j$. Since we can easily compute the metric's optimal rate over $\mathcal{S}$ using Lemma 1, by varying $\boldsymbol{\theta}$ in this procedure, we parametrize the surface of the sphere $\partial\mathcal{S}$. We denote this parametrization by $\mu(\boldsymbol{\theta})$, where $\mu : [0,\pi]^{k-2} \times [0, 2\pi] \to \partial\mathcal{S}$.

Description of Algorithm 2: Let the oracle's metric be $\phi^{\text{lin}} = \langle \mathbf{a}, \mathbf{r} \rangle$ such that $\|\mathbf{a}\|_2 = 1$ (Section 2.2). Using the parametrization $\mu(\boldsymbol{\theta})$ for the boundary of the sphere $\partial\mathcal{S}$, Algorithm 2 returns an estimate $\hat{\mathbf{a}}$ with $\|\hat{\mathbf{a}}\|_2 = 1$. Line 2-6 recover the search orthant of the optimal rate over the sphere by posing k trivial queries. Once the search orthant of the optimal rate is known, the algorithm in each iteration of the for loop (line 9-18) updates one angle θ_j keeping other angles fixed by the *ShrinkInterval* subroutine. The *ShrinkInterval* subroutine (illustrated in Figure 5) is binary-search based routine that shrinks the interval $[\theta_j^a, \theta_j^b]$ by half based on the oracle responses to (at most) three queries.[1] Note that, depending on the oracle responses, one may reduce the search interval to half using less than three queries in some cases. Then the algorithm cyclically updates each angle until it converges to a metric sufficiently close to the true metric. We fix the number of cycles in coordinate-wise binary search to three. Therefore, in order to elicit a linear performance metric in k dimensions, the LPME subroutine requires at most $3 \times 3 \times k \log(\pi/2\epsilon)$ queries, where three is the number of cycles in coordinate wise binary search, three is the (maximum) number of queries to shrink the search interval into half, and the initial search interval for the angles is $\pi/2$.

B GEOMETRY OF THE FEASIBLE SPACE (PROOFS OF SECTION 2 AND SECTION 4)

Proof of Proposition 1 and Proposition 2. We prove Proposition 2. The proof of Proposition 1 is analogous where the probability measures (corresponding to classifiers and their rates) are not conditioned on any group.

The group-specific set of rates $\mathcal{R}^g$ for a group g has the following properties [Hiranandani et al., 2020]:

- *Convex*: Consider two classifiers $h_1^g, h_2^g \in \mathcal{H}^g$ that achieve the rates $\mathbf{r}_1^g, \mathbf{r}_2^g \in \mathcal{R}^g$, respectively. Also, consider a classifier h^g that predicts what classifier h_1^g predicts with probability γ and predicts what classifier h_2^g predicts with probability $1-\gamma$. Then the rate vector of the classifier h^g is:
$$\begin{aligned} R_{ij}^g(h) &= \mathbb{P}(h^g = i | Y = i) \\ &= \mathbb{P}(h_1^g = i | h^g = h_1^g, Y = i)\mathbb{P}(h^g = h_1^g) + \mathbb{P}(h_2^g = i | h^g = h_2^g, Y = i)\mathbb{P}(h^g = h_2^g) \\ &= \gamma \mathbf{r}_1^g + (1-\gamma)\mathbf{r}_2^g. \end{aligned}$$
The above equations show that the convex combination of any two rates is feasible as well, i.e., one can construct a randomized classifier which will achieve the convex combination of rates. Hence, $\mathcal{R}^g \;\forall\; g \in [m]$ is convex. Since intersection of convex sets is convex, the intersection set $\mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$ is convex as well.
- *Bounded:* Since $R_{ij}^g(h) = P[h = i | Y = i] \leq 1$ for all $i \in [k]$, $\mathcal{R}^g \subseteq [0,1]^k$.
- *The rates* $\mathbf{o}$ *and* $\mathbf{e}_i$*'s are always achievable:* A uniform random classifier, i.e, the classifier, which for any input, predicts all classes with probability $1/k$ achieves the rate profile $\mathbf{o}$. A classifier that always predicts class i achieves the rate $\mathbf{e}_i$. Thus, $\mathbf{e}_i \in \mathcal{R}^g \,\forall\, i \in [k], g \in [m]$ are always feasible.

[1] The description of the binary search algorithm in Hiranandani et al. [2019b] always assumes getting responses to four queries that essentially correspond to the four intervals. In practice, the binary search can be adaptive and may only require at most three queries as we have discussed in this paper. The order of the queries in LPME, however, remains the same.

Algorithm 3 Obtaining the sphere $\mathcal{S} \subset \mathcal{R}$ (Figure 2(a)) of radius ρ centered at $\mathbf{o}$

1: **for** $j = 1, 2, \cdots, k$ **do**
2: Let $\boldsymbol{\alpha}_j$ be the standard basis vector.
3: Compute the maximum constant c_j such that $\mathbf{o} + c_j \boldsymbol{\alpha}_j$ is feasible by solving (OP1).
4: Let $CONV$ denote the convex hull of $\{\mathbf{o} \pm c_j \boldsymbol{\alpha}_j\}_{j=1}^{k}$. It will be centered at $\mathbf{o}$.
5: Compute the radius ρ of the largest ball that fits in $CONV$.
6: **Output:** Sphere $\mathcal{S}$ with radius ρ centered at $\mathbf{o}$.

- $\mathbf{e}_i$*'s are vertices:* Consider the supporting hyperplanes with the following slope: $a_i > a_j > 0$ and $a_l = 0$ for $l \in [k], l \neq i, j$. These hyperplanes will be supported by $\mathbf{e}_i$. Thus, $\mathbf{e}_i$'s are vertices of the convex set $\mathcal{R}^g$. From Assumption 1, one can construct a ball around the trivial rate $\mathbf{o}$ and thus $\mathbf{o}$ lies in the interior.

The above points apply to space of overall rates $\mathcal{R}$ as well; thus, proving Proposition 1. □

B.1 FINDING THE SPHERE $\mathcal{S} \subset \mathcal{R}$

In this section, we provide details regarding how a sphere $\mathcal{S}$ with sufficiently large radius ρ inside the feasible region $\mathcal{R}$ may be found (see Figure 2(a)). The following discussion is borrowed from [Hiranandani et al., 2019b] and provided here for completeness.

The following optimization problem is a special case of OP2 in [Narasimhan, 2018]. The problem is associated with a feasibility check problem. Given a rate profile $\mathbf{r}_0$, the optimization routine tries to construct a classifier that achieves the rate $\mathbf{r}_0$ within small error $\epsilon > 0$.

$$\min_{\mathbf{r} \in \mathcal{R}} \; 0 \qquad s.t. \;\; \|\mathbf{r} - \mathbf{r}_0\|_2 \leq \epsilon. \tag{OP1}$$

The above optimization problem checks the feasibility, and if there exists a solution to the above problem, then Algorithm 1 of [Narasimhan, 2018] returns it. Furthermore, Algorithm 3 computes a value of $\rho \geq \tilde{p}/k$, where $\tilde{p}$ is the radius of the largest ball contained in the set $\mathcal{R}$. Also, the approach in [Narasimhan, 2018] is consistent, thus we should get a good estimate of the sphere, provided we have sufficiently large number of samples. The algorithm is completely offline and does not impact oracle query complexity.

Lemma 2. *[Hiranandani et al., 2019b] Let $\tilde{p}$ denote the radius of the largest ball in $\mathcal{R}$ centered at $\mathbf{o}$. Then Algorithm 3 returns a sphere with radius $\rho \geq \tilde{p}/k$, where k is the number of classes.*

The idea in Algorithm 3 can be trivially extended to finding a sphere $\overline{S} \subset \mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$ corresponding to Remark 3.

C QUADRATIC PERFORMANCE METRIC ELICITATION PROCEDURE

In this section, we describe how the subroutine calls to LPME in Algorithm 1 elicit a quadratic metric in Definition 3. We start with the shifted metric of Equation (5). Also, as explained in the main paper, we may assume $d_1 \neq 0$ due to Assumption 2. We can derive the following solution using any non-zero coordinate of $\mathbf{d}$, instead of d_1. We can identify a non-zero coordinate using k trivial queries of the form $(\varrho \boldsymbol{\alpha}_i + \mathbf{o}, \mathbf{o}), \forall i \in [k]$.

1. From line 1 of Algorithm 1, we get local linear approximation at $\mathbf{o}$. Using Remark 2, we have (6) which is

$$d_i = \frac{f_{i0}}{f_{10}} d_1 \qquad \forall \, i \in \{2, \ldots, k\}. \tag{13}$$

2. Similarly, if we apply LPME on small balls around rate profiles $\mathbf{z}_j$, Remark 2 gives us:

$$\frac{d_i + (\rho - \varrho) B_{ij}}{d_1 + (\rho - \varrho) B_{1j}} = \frac{f_{ij}}{f_{1j}} \quad \forall \, i \in \{2, \ldots, k\}, \; j \leq i. \tag{14}$$

$$
\begin{aligned}
&\implies d_i + (\rho - \varrho)B_{ij} = \frac{f_{ij}}{f_{1j}}(d_1 + (\rho - \varrho)B_{1j}) \\
&\implies (\rho - \varrho)B_{ij} = \frac{f_{ij}}{f_{1j}}(d_1 + (\rho - \varrho)B_{j1}) - d_i \\
&\implies (\rho - \varrho)B_{ij} = \frac{f_{ij}}{f_{1j}}(d_1 + \frac{f_{j1}}{f_{11}}(d_1 + (\rho - \varrho)B_{11}) - d_j) - \frac{f_{i0}}{f_{10}}d_1 \\
&\implies (\rho - \varrho)B_{ij} = \left(\frac{f_{ij}}{f_{1j}} - \frac{f_{i0}}{f_{10}} + \frac{f_{ij}}{f_{1j}}\left(\frac{f_{j1}}{f_{11}} - \frac{f_{j0}}{f_{10}}\right)\right)d_1 + (\rho - \varrho)\frac{f_{j1}}{f_{11}}B_{11},
\end{aligned} \tag{15}
$$

where we have used that the matrix $\mathbf{B}$ is symmetric in the second step, and (13) in the last two steps. We can represent each element in terms of B_{11} and d_1. So, a relation between B_{11} and d_1 may allow us to represent each element of $\mathbf{a}$ and $\mathbf{B}$ in terms of d_1.

3. Therefore, by applying LPME on small balls around rate profiles $-\mathbf{z}_1$, Remark 2 gives us (8):

$$
\frac{d_2 - (\rho - \varrho)B_{21}}{d_1 - (\rho - \varrho)B_{11}} = \frac{f_{21}^-}{f_{11}^-}. \tag{16}
$$

4. Using (14) and (16), we have:

$$
(\rho - \varrho)B_{11} = \frac{\frac{f_{21}^-}{f_{11}^-} + \frac{f_{21}}{f_{11}} - 2\frac{f_{20}}{f_{10}}}{\frac{f_{21}^-}{f_{11}^-} - \frac{f_{21}}{f_{11}}}d_1. \tag{17}
$$

Putting (17) in (15), we get:

$$
\begin{aligned}
B_{ij} &= \left[\frac{f_{ij}}{f_{1j}}\left(1 + \frac{f_{j1}}{f_{11}}\right) - \frac{f_{ij}}{f_{1j}}\frac{f_{j0}}{f_{10}} - \frac{f_{i0}}{f_{10}} + \frac{f_{ij}}{f_{1j}}\frac{f_{j1}}{f_{11}}\frac{\frac{f_{21}^-}{f_{11}^-} + \frac{f_{21}}{f_{11}} - 2\frac{f_{20}}{f_{10}}}{\frac{f_{21}^-}{f_{11}^-} - \frac{f_{21}}{f_{11}}}\right]d_1 \\
&= \left(F_{i,1,j}(1 + F_{j,1,1}) - F_{i,1,j}F_{j,1,0} - F_{i,1,0} + F_{i,1,j}\frac{F_{2,1,1}^- + F_{2,1,1} - 2F_{2,1,0}}{F_{2,1,1}^- - F_{2,1,1}}\right)d_1,
\end{aligned} \tag{18}
$$

where $F_{i,j,l} = \frac{f_{il}}{f_{jl}}$ and $F_{i,j,l}^- = \frac{f_{il}^-}{f_{jl}^-}$. As $\mathbf{a} = \mathbf{d} + \mathbf{B}\mathbf{o}$, we can represent each element of $\mathbf{a}$ and $\mathbf{B}$ using using (13) and (18) in terms of d_1. We can then use the normalization condition $\|\mathbf{a}\|_2^2 + \|\mathbf{B}\|_F^2 = 1$ to get estimates of $\mathbf{a}, \mathbf{B}$ which are independent of d_1.

This completes the derivation of solution from QPME (section 3).

D FAIR (QUADRATIC) PERFORMANCE METRIC ELICITATION PROCEDURE

Algorithm 4: FPM Elicitation

Input: Query set $\mathcal{S}'$, search tolerance $\epsilon > 0$, oracle Ω'

1. Let $\mathcal{L} \leftarrow \varnothing$
2. **For** $\sigma \in \mathcal{M}$ **do**
3. $\quad \beta^\sigma \leftarrow$ QPME$(\mathcal{S}', \epsilon, \Omega')$
4. $\quad$ Let ℓ^σ be Eq. (22), extend $\mathcal{L} \leftarrow \mathcal{L} \cup \{\ell^\sigma\}$
5. $\hat{\mathbb{B}} \leftarrow$ normalized solution from (25) using $\mathcal{L}$
6. $\hat{\lambda} \leftarrow$ trace back normalized solution from (22) for any σ

Output: $\hat{\mathbf{a}}, \hat{\mathbb{B}}, \hat{\lambda}$

We first discuss eliciting the fair (quadratic) metric in Definition 4, where all the parameters are unknown. We then provide an alternate procedure for eliciting just the trade-off parameter λ when the predictive performance and fairness violation

coefficients are known. The latter is a separate application as discussed in [Zhang et al., 2020]. However, unlike Zhang et al. [2020], instead of ratio queries, we use simpler pairwise comparison queries.

In this section, we work with any number of groups $m \geq 2$. The idea, however, remains the same as described in the main paper for number of groups $m = 2$. We specifically select queries from the sphere $\overline{\mathcal{S}} \subset \mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$, which is common to all the group-specific feasible region of rates, so to reduce the problem into multiple instances of the proposed QPME procedure of Section 3.

Suppose that the oracle's fair performance metric is ϕ^{fair} parametrized by $(\mathbf{a}, \mathbb{B}, \lambda)$ as in Definition 4. The overall fair metric elicitation procedure framework is summarized in Algorithm 4. The framework exploits the sphere $\overline{\mathcal{S}} \subset \mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$ and uses the QPME procedure (Algorithm 1) as a subroutine multiple times.

Let us consider a non-empty set of sets $\mathcal{M} \subset 2^{[m]} \setminus \{\varnothing, [m]\}$. We will later discuss how to choose such a set $\mathcal{M}$. We partition the set of groups $[m]$ into two sets of groups. Let $\sigma \in \mathcal{M}$ and $[m] \setminus \sigma$ be one such partition of the m groups defined by the set of groups σ. For example, when $m = 3$, one may choose the set of groups $\sigma = \{1, 2\}$.

Now, consider a sphere $\mathcal{S}'$ whose elements $\mathbf{r}^{1:m} \in \mathcal{S}'$ are given by:

$$\mathbf{r}^g = \begin{cases} \mathbf{s} & \text{if } g \in \sigma \\ \mathbf{o} & \text{o.w.} \end{cases} \tag{19}$$

This is an extension of the sphere $\mathcal{S}'$ defined in the main paper for the $m > 2$ case. Elements in $\mathcal{S}'$ have rate profiles $\mathbf{s} \in \overline{\mathcal{S}}$ to the groups in σ and trivial rate profile $\mathbf{o}$ to the remaining groups in $[m] \setminus \sigma$. Analogously, the modified oracle is $\Omega'(\mathbf{r}_1, \mathbf{r}_2) = \Omega((\mathbf{r}_1^{1:m}), (\mathbf{r}_2^{1:m}))$, where $\mathbf{r}_1^{1:m}, \mathbf{r}_2^{1:m}$ are the elements of the spheres $\mathcal{S}'$ above. Thus, for elements in $\mathcal{S}'$, the metric in Definition 4 reduces to:

$$\phi^{\text{fair}}(\mathbf{r}^{1:m} \in \mathcal{S}'\,; \mathbf{a}, \mathbb{B}, \lambda) = (1-\lambda)\langle -\mathbf{a} \odot \boldsymbol{\tau}^\sigma, \mathbf{s} - \mathbf{o}\rangle + \lambda \frac{1}{2}(\mathbf{s} - \mathbf{o})^T \mathbf{W}^\sigma (\mathbf{s} - \mathbf{o}) + c^\sigma \tag{20}$$

where $\boldsymbol{\tau}^\sigma = \sum_{g \in \sigma} \boldsymbol{\tau}^g$, $\mathbf{W}^\sigma = \sum_{u \in \sigma, v \in [m] \setminus \sigma} B^{uv}$, and c^σ is a constant not affecting the oracle responses.

The above metric is a particular instance of $\overline{\phi}(\mathbf{s}; \mathbf{d}, \mathbf{B})$ in (4) with $\mathbf{d} := -(1-\lambda)\mathbf{a} \odot \boldsymbol{\tau}^\sigma$ and $\mathbf{B} := \lambda \mathbf{W}^\sigma$; thus, we apply QPME procedure as a subroutine in Algorithm 4 to elicit the metric in (20).

The only change needed to be made to the algorithm is in line 5, where we need to take into account the changed relationship between $\mathbf{d}$ and $\mathbf{a}$, and need to separately (not jointly) normalize the linear and quadratic coefficients. With this change, the output of the algorithm directly gives us the required estimates. Specifically, we have from step 1 of Algorithm 1 and (6) an estimate

$$\frac{d_i}{d_1} = \frac{\tau_i^\sigma a_i}{\tau_1^\sigma a_1} = \frac{f_{i0}}{f_{10}} \implies a_i = \frac{f_{i0}}{f_{10}} \frac{\tau_1^\sigma}{\tau_i^\sigma} a_1. \tag{21}$$

Using the normalization condition (i.e., $\|\mathbf{a}\|_2 = 1$), we directly get an estimate $\hat{\mathbf{a}}$ for the linear coefficients. Similarly, steps 2-4 of Algorithm 1 and (9) gives us:$\hat{B}_{ij} =$

$$\begin{aligned} \sum_{u \in \sigma, v \in [m] \setminus \sigma} \tilde{B}_{ij}^{uv} &= \Big(F_{i,1,j}^\sigma (1 + F_{j,1,1}^\sigma) - F_{i,1,j}^\sigma F_{j,1,0}^\sigma d_1 - F_{i,1,0}^\sigma + F_{i,1,j}^\sigma \tfrac{F_{2,1,1}^{-,\sigma} + F_{2,1,1}^\sigma - 2F_{2,1,0}^\sigma}{F_{2,1,1}^{-,\sigma} - F_{2,1,1}^\sigma} \Big) \tau_1^1 \hat{a}_1 \\ &= \beta^\sigma, \end{aligned} \tag{22}$$

where the above solution is similar to the two group case, but here it is corresponding to a partition of groups defined by σ, and $\tilde{\mathbf{B}}^{uv} := \lambda \mathbf{B}^{uv}/(1-\lambda)$ is a scaled version of the true (unknown) $\mathbf{B}^{uv}$. Let equation (22) be denoted by ℓ^σ. Also, let the right hand side term of (22) be denoted by β^σ.

Since we want to elicit $\binom{m}{2}$ fairness violation weight matrices in $\mathbb{B}$, we require $\binom{m}{2}$ ways of partitioning the groups into two sets so that we construct $\binom{m}{2}$ independent matrix equations similar to (22). Let $\mathcal{M}$ be those set of sets. Thus, running over all the choices of sets of groups $\sigma \in \mathcal{M}$ provides the system of equations $\mathcal{L} := \cup_{\sigma \in \mathcal{M}} \ell^\sigma$ (line 4 in Algorithm 4), which is:

$$\begin{bmatrix} \Xi & 0 & \dots & 0 \\ 0 & \Xi & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \Xi \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{b}}_{(11)} \\ \tilde{\mathbf{b}}_{(12)} \\ \dots \\ \tilde{\mathbf{b}}_{(kk)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\beta}_{(11)} \\ \boldsymbol{\beta}_{(12)} \\ \dots \\ \boldsymbol{\beta}_{(kk)} \end{bmatrix}, \tag{23}$$

where $\tilde{\mathbf{b}}_{(ij)} = (\tilde{b}^1_{ij}, \tilde{b}^2_{ij}, \cdots, \tilde{b}^{\binom{m}{2}}_{ij})$ and $\boldsymbol{\gamma}_{(ij)} = (\beta^1_{ij}, \beta^2_{ij}, \cdots, \beta^{\binom{m}{2}}_{ij})$ are vectorized versions of the ij-th entry across groups for $i, j \in [k]$, and $\Xi \in \{0,1\}^{\binom{m}{2}\times\binom{m}{2}}$ is a binary full-rank matrix denoting membership of groups in the set σ. For example, when one chooses $\mathcal{M} = \{\{1,2\},\{1,3\},\{2,3\}\}$ for $m = 3$, Ξ is given by:

$$\Xi = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

One may choose any set of sets $\mathcal{M}$ that allows the resulting group membership matrix Ξ to be non-singular. The solution of the system of equations $\mathcal{L}$ is:

$$\begin{bmatrix} \tilde{\mathbf{b}}_{(11)} \\ \tilde{\mathbf{b}}_{(12)} \\ \cdots \\ \tilde{\mathbf{b}}_{(kk)} \end{bmatrix} = \begin{bmatrix} \Xi & 0 & \dots & 0 \\ 0 & \Xi & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \Xi \end{bmatrix}^{(-1)} \begin{bmatrix} \boldsymbol{\beta}_{(11)} \\ \boldsymbol{\beta}_{(12)} \\ \cdots \\ \boldsymbol{\beta}_{(kk)} \end{bmatrix}. \tag{24}$$

When all $\tilde{\mathbf{B}}^{uv}$'s are normalized, we have the estimated fairness violation weight matrices as:

$$\hat{\mathbf{B}}^{uv} = \frac{\tilde{\mathbf{B}}^{uv}}{\frac{1}{2}\sum_{u,v=1,v>u}^{m} \|\tilde{\mathbf{B}}^{uv}\|_F} \quad \text{for} \quad u, v \in [m], v > u. \tag{25}$$

Due to the above normalization, the solution is again independent of the true trade-off λ.

Given estimates $\hat{B}^{uv}_{ij}$ and $\hat{a}_1$, we can now additionally estimate the trade-off parameter $\hat{\lambda}$ from ℓ^σ (22) for any $\sigma \in \mathcal{M}$. This completes the fair (quadratic) metric elicitation procedure.

D.1 ELICITING TRADE-OFF λ WHEN (LINEAR) PREDICTIVE PERFORMANCE AND (QUADRATIC) FAIRNESS VIOLATION COEFFICIENTS ARE KNOWN

We now provide an alternate binary search based method similar to Hiranandani et al. [2020] for eliciting the trade-off parameter λ when the linear predictive and quadratic fairness coefficients are already known. This is along similar lines to the application considered by Zhang et al. [2020], but unlike them, instead of ratio queries, we require simpler pairwise queries.

Here, the key insight is to approximate the non-linearity posed by the fairness violation in Definition 4, which then reduces the problem to a one-dimensional binary search. We have:

$$\phi^{\text{fair}}(\mathbf{r}^{1:m}\,;\,\mathbf{a}, \mathbb{B}, \lambda) \coloneqq (1-\lambda)\langle \mathbf{a}, \mathbf{1}-\mathbf{r}\rangle + \lambda\frac{1}{2}\left(\sum\nolimits_{u,v=1,v>u}^{m} (\mathbf{r}^u - \mathbf{r}^v)^T \mathbf{B}^{uv}(\mathbf{r}^u - \mathbf{r}^v)\right). \tag{26}$$

To this end, we define a new sphere $\mathcal{S}' = \{(\mathbf{s}, \mathbf{o}, \dots, \mathbf{o}) | \mathbf{s} \in \overline{\mathcal{S}}\}$. The elements in $\mathcal{S}'$ is the set of rate profiles whose first group achieves rates $\mathbf{s} \in \overline{\mathcal{S}}$ and rest of the groups achieve trivial rate $\mathbf{o}$ (corresponding to uniform random classifier). For any element in $\mathcal{S}'$, the associated discrepancy terms $(\mathbf{r}^u - \mathbf{r}^v) = 0$ for $u, v \neq 1$. Thus for elements in $\mathcal{S}'$, the metric in Definition 4 reduces to:

$$\phi^{\text{fair}}((\mathbf{s}, \mathbf{o}, \dots, \mathbf{o})\,;\,\mathbf{a}, \mathbb{B}, \lambda) = (1-\lambda)\langle -\boldsymbol{\tau}^1 \odot \mathbf{a}, \mathbf{s}-\mathbf{o}\rangle + \lambda\frac{1}{2}(\mathbf{s}-\mathbf{o})^T \sum_{v=2}^{m} \mathbf{B}^{1v}(\mathbf{s}-\mathbf{o}) + c. \tag{27}$$

Additionally, we consider a small sphere $\overline{\mathcal{S}}'_{\mathbf{z}_1}$, where $\mathbf{z}_1 \coloneqq (\rho - \varrho)\boldsymbol{\alpha}_1 + \mathbf{o}$, similar to what is shown in Figure 2(a). We may approximate the quadratic term on the right hand side above by its first order Taylor approximation as follows:

$$\begin{aligned} \phi^{\text{fair}}((\mathbf{s}, \mathbf{o}, \dots, \mathbf{o}); \mathbf{a}, \mathbb{B}, \lambda) &\approx \phi^{\text{fair, apx}}((\mathbf{s}, \mathbf{o}, \dots, \mathbf{o}); \mathbf{a}, \mathbb{B}, \lambda) \\ &= \langle -(1-\lambda)\boldsymbol{\tau}^1 \odot \mathbf{a} + \lambda \sum_{v=2}^{m} \mathbf{B}^{1v}(\mathbf{z}_1 - \mathbf{o}), \mathbf{s}\rangle \end{aligned} \tag{28}$$

for $\mathbf{s}$ in a small neighbourhood around the rate profile $\mathbf{z}_1$. Since the metric is essentially linear in $\mathbf{s}$, the following lemma from [Hiranandani et al., 2020] shows that the metric in (28) is quasiconcave in λ.

Algorithm 5 Eliciting the trade-off λ when predictive performance and fairness violation are known

1: **Input:** Query space $\overline{\mathcal{S}}'_{\mathbf{z}_1}$, binary-search tolerance $\epsilon > 0$, oracle Ω
2: **Initialize:** $\lambda^{(a)} = 0$, $\lambda^{(b)} = 1$.
3: **while** $\left|\lambda^{(b)} - \lambda^{(a)}\right| > \epsilon$ **do**
4: Set $\lambda^{(c)} = \frac{3\lambda^{(a)}+\lambda^{(b)}}{4}$, $\lambda^{(d)} = \frac{\lambda^{(a)}+\lambda^{(b)}}{2}$, $\lambda^{(e)} = \frac{\lambda^{(a)}+3\lambda^{(b)}}{4}$
5: Set $\mathbf{s}^{(a)} = \underset{\mathbf{s}\in\overline{\mathcal{S}}'_{\mathbf{z}_1}}{\operatorname{argmax}} \langle -(1-\lambda^{(a)})\boldsymbol{\tau}^1 \odot \hat{\mathbf{a}} + \lambda^{(a)} \sum_{v=2}^{m} \hat{\mathbf{B}}^{1v}(\mathbf{z}_1 - \mathbf{o}), \mathbf{s}\rangle$ using Lemma 1
6: Similarly, set $\mathbf{s}^{(c)}$, $\mathbf{s}^{(d)}$, $\mathbf{s}^{(e)}$, $\mathbf{s}^{(b)}$.
7: $[\lambda^{(a)}, \lambda^{(b)}] \leftarrow$ *ShrinkInterval* $(\Omega, \mathbf{s}^{(a)}), \mathbf{s}^{(c)}), \mathbf{s}^{(d)}), \mathbf{s}^{(e)}), \mathbf{s}^{(b)}))$ using a subroutine analogous to the routine shown in Figure 5.
8: **Output:** $\hat{\lambda} = \frac{\lambda^{(a)}+\lambda^{(b)}}{2}$.

Lemma 3. *Under the regularity assumption that* $\langle -\boldsymbol{\tau}^1 \odot \mathbf{a}, \sum_{v=2}^{m} \mathbf{B}^{1v}(\mathbf{z}_1 - \mathbf{o})\rangle \neq 1$, *the function*

$$\vartheta(\lambda) := \max_{\mathbf{s}\in\overline{\mathcal{S}}'_{\mathbf{z}_1}} \phi^{\textit{fair, apx}}((\mathbf{s}, \mathbf{o}, \ldots, \mathbf{o}); \mathbf{a}, \mathbb{B}, \lambda) \tag{29}$$

is strictly quasiconcave (and therefore unimodal) in λ.

The unimodality of $\vartheta(\lambda)$ allows us to perform the one-dimensional binary search in Algorithm 5 using the query space $\overline{\mathcal{S}}'_{\mathbf{z}_1}$, tolerance ϵ, and the oracle Ω. The binary search algorithm is same as Algorithm 4 in [Hiranandani et al., 2020] and provided here for completeness.

E EXTENSION TO ELICITING GENERAL QUADRATIC METRICS

In this section, we discuss how the entire setup including the proposed procedure and the guarantees of the main paper described in terms of the *diagonal* entries of the predictive rate matrix extends to a setup where the metric is defined in all the terms of the rate matrix. For this section, we use an additional notation. For a matrix $\mathbf{A}$, let $\textit{off-diag}(\mathbf{A})$ returns a vector of off-diagonal elements of $\mathbf{A}$.

Just like the main paper, we consider a k-class classification setting with $X \in \mathcal{X}$ and $Y \in [k]$ denoting the input and output random variables, respectively. We assume access to an n-sized sample $\{(\mathbf{x}, y)_i\}_{i=1}^n$ generated *iid* from a distribution $\mathbb{P}(X, Y)$. We work with randomized classifiers $h : \mathcal{X} \to \Delta_k$ that for any $\mathbf{x}$ gives a distribution $h(\mathbf{x})$ over the k classes and use $\mathcal{H} = \{h : \mathcal{X} \to \Delta_k\}$ to denote the set of all classifiers.

General Predictive rates: We define the predictive rate matrix for a classifier h by $\mathbf{R}(h, \mathbb{P}) \in \mathbb{R}^{k\times k}$, where the ij-th entry is the fraction of label-i examples for which the randomized classifier h predicts j:

$$R_{ij}(h, \mathbb{P}) := P(h(X) = j | Y = i) \quad \text{for } i, j \in [k], \tag{30}$$

where the probability is over draw of $(X, Y) \sim \mathbb{P}$ and the randomness in h.

Notice that each diagonal entry of $\mathbf{R}$ can be written in terms of its off-diagonal elements:

$$R_{ii}(h, \mathbb{P}) = 1 - \sum\nolimits_{j=1, j\neq i}^{k} R_{ij}(h, \mathbb{P}).$$

Thus, we can represent a rate matrix with its $q := (k^2 - k)$ off-diagonal elements, write it as a vector $\mathbf{r}(h, \mathbb{P}) = \textit{off-diag}(\mathbf{R}(h, \mathbb{P}))$, and interchangeably refer to it as the *'vector of general rates'* or *'off-diagonal rates'*. To distinguish from rates considered in the main paper, we will call the rates entries corresponding to the diagonals of the rate matrix, i.e., $\mathbb{P}(h(X) = i | Y = i)$ as discussed in Equation(1), as the *'diagonal rates'*.

Feasible general rates: The set of all feasible general rates is given by:

$$\mathcal{R} = \{\mathbf{r}(h, \mathbb{P}) \in [0, 1]^q \,:\, h \in \mathcal{H}\}.$$

The quadratic metric in general rates is defined in the same way as Definition 3 as follows:

Definition 6 (Quadratic Metric in General Rates)**.** *For a vector* $\mathbf{a} \in \mathbb{R}^q$ *and a positive semi-definite symmetric matrix* $\mathbf{B} \in \mathbb{R}^{q \times q}$ *with* $\|\mathbf{a}\|_2^2 + \|\mathbf{B}\|_F^2 = 1$ *(w.l.o.g. due to scale invariance):*

$$\phi^{\text{quad}}(\mathbf{r}\,;\,\mathbf{a}, \mathbf{B}) = \langle \mathbf{a}, \mathbf{r} \rangle + \frac{1}{2}\mathbf{r}^T \mathbf{B} \mathbf{r}. \tag{31}$$

Example 4 (Distribution matching)**.** We can extend Example 3 in the multiclass case as follows. In certain applications, one needs the proportion of predictions for each class (i.e., the coverage) to match a target distribution $\boldsymbol{\pi} \in \Delta_k$ [Goh et al., 2016, Narasimhan, 2018]. A measure often used for this task is the squared difference between the per-class coverage and the target distribution: $\phi^{\text{cov}}(\mathbf{r}) = \sum_{i=1}^{k} \left(\text{cov}_i(\mathbf{r}) - \pi_i\right)^2$, where $\text{cov}_i(\mathbf{r}) = 1 - \sum_{j=1}^{k-1} r_{(i-1)(k-1)+j} + \sum_{j>i} r_{(j-1)(k-1)+i} + \sum_{j<i} r_{(j-1)(k-1)+i-1}$. Similar metrics can be found in the quantification literature where the target is set to the class prior $\mathbb{P}(Y = i)$ [Esuli and Sebastiani, 2015, Kar et al., 2016]. We capture more general quadratic distance measures for distributions, e.g. $(\text{cov}(\mathbf{r}) - \boldsymbol{\pi})^{\mathbf{T}} \mathbf{Q} (\text{cov}(\mathbf{r}) - \boldsymbol{\pi})$ for a positive semi-definite matrix $\mathbf{Q} \in PSD_k$ [Lindsay et al., 2008].

The definition of metric elicitation and oracle query remain the same except that the vector $\mathbf{r}$ now represents the vector of general rates, and not just the diagonal rates.

Consider Appendix B, where we discuss proof of Proposition 1, Proposition 2, and a procedure to construct a feasible sphere of appropriate radius in the convex set of diagonal rates. The entire methodology applies to general set of rates by *just* replacing diagonal rates in the proofs with the general rates. Thus, all the geometrical properties discussed in Proposition 1 for the set of diagonal rates applies to the set of general rates. The exact geometry of the set of diagonal rates, as shown in Figures 2(a) and 2(b)may differ from the geometry of the set of general rates; however, the geometric properties including $\mathbf{e}_i$ being the vertices remains the same. Therefore, under the same Assumption 1, we can guarantee an existence of a sphere in the set of general rates similar to Remark 1.

Once we guarantee a sphere in the set of general rates, we can follow LPME for eliciting linear metrics in general rates or QPME for eliciting linear metrics in general rates. The computational and query complexity will depend on the number of unknowns, which in the case of general rates, will be $\tilde{O}(q)$ for LPME and $\tilde{O}(q^2)$ for QPME.

F PRACTICALITY OF QUERYING ORACLE

Recall that in our setup, any query posed to the oracle needs to feasible, i.e., should be achievable by some classifier (see definition of feasible rates in Section 2). Therefore the oracle we query can be a human expert or a group of experts who compare intuitive visualizations of rates, or can be an entire population of users (as would be the case with A/B testing).

An important practical concern in employing the proposed QPME procedure is the number of queries needed to be posed to the oracle. We note that (i) the number of queries needed by our proposal is optimal (i.e. matches the lower bound for the problem in Theorem 2), (ii) has only a *linear* dependence on the number of unknowns (Theorem 1), and (iii) can be considerably reduced by making reasonable practical structural assumptions about the metric to reduce the number of unknowns. While our procedure's query complexity for the most general setup (with $O(k^2)$ unknowns) is $\tilde{O}(k^2)$, the quadratic dependence on k is merely an artifact of there being $O(k^2)$ unknowns in this setup. For example, when the number of classes is large, one may just cluster the classes from error perspective. For example, one may assume same error costs for similar classes. This will reduce the number of unknowns to $O(c^2)$, where $c << k$ is the number of cluster of classes.

We also stress that in many internet-based settings, one can deploy A/B tests to obtain preferences by aggregating feedback from a large group of participants. In this case, the entire user population serves as an oracle. Most internet-based companies run thousands of A/B tests daily making it practical to get preferences for our metric elicitation procedure. Moreover, one can employ practical improvements such as running A/B tests with fewer participants in the initial rounds (when the rates are far apart) and switch to running A/B tests with more precision in later rounds. Note that because the queries posed by our method always corresponds to a feasible classifier (see Section 3), one can easily run comparisons between classifiers as a part of an A/B test.

If needed, our algorithms can also work with queries that compare classification statistics directly, instead of classifiers. There has been growing work on visualization of confusion matrices (predictive rates) for non-expert users. For example, see [Beauxis-Aussalet and Hardman, 2014] and [Shen et al., 2020]. With the aid of such intuitive visualizations, it is

reasonable to expect human practitioners to comprehend the queries posed to them and provide us with pairwise comparisons. Moreover, we have shown that our approach is resilient to noisy responses, which enhances our confidence in their ability to handle human feedback. In practice, the most viable option will depend on the target population.

Finally, we would like to emphasize that because our query complexity has only a linear dependence on the number of unknowns, and the number of unknowns can be reduced with practical structural assumptions, our proposal is as practical as the prior methods [Hiranandani et al., 2019a,b] for linear metric elicitation. In fact, despite eliciting from a more flexible class, our proposal has the same dependence on the number of unknowns as those prior methods.

G ELICITATION GUARANTEE FOR THE QPME PROCEDURE

G.1 SAMPLE COMPLEXITY BOUNDS

Recall from Definition 5 that the oracle responds correctly as long as $|\phi(\mathbf{r}_1) - \phi(\mathbf{r}_2)| > \epsilon_\Omega$. For simplicity, we assume that our algorithm has access to the population rates $\mathbf{r}$ defined in Eq. (1). In practice, we expect to estimate the rates using a sample $D := \{\mathbf{x}, y\}_{i=1}^n$ drawn from the distribution $\mathbb{P}$, and to query classifiers from a hypothesis class $\mathcal{H}$ with finite capacity. Standard generalization bounds (e.g. Daniely et al. [2015]) give us that with high probability over draw of D, the estimates $\hat{\mathbf{r}}$ are close to the population rates $\mathbf{r}$, up to the desired tolerance ϵ_Ω, as long as we have sufficient samples. Further, since the metrics ϕ are Lipschitz w.r.t. rates, with high probability, we thus gather correct oracle feedback from querying with finite sample estimates $\Omega(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2)$.

More formally, for $\delta \in (0, 1)$, as long as the sample size n is greater than $O\left(\log(|\mathcal{H}|/\delta)/\epsilon_\Omega^2\right)$, the guarantee in Theorem 1 hold with probability at least $1 - \delta$ (over draw of D), where $|\mathcal{H}|$ can in turn be replaced by a measure of capacity of the hypothesis class $\mathcal{H}$. For example, one can show the following corollary to Theorem 1 for a hypothesis class $\mathcal{H}$ in which each classifier is a randomized combination of a finite number of deterministic classifiers chosen from a set $\bar{\mathcal{H}}$, and whose capacity is measured in terms of the Natarajan dimension [Natarajan, 1989] of $\bar{\mathcal{H}}$.

Corollary 1. *Suppose the hypothesis class $\mathcal{H}$ of randomized classifiers used to choose queries to the oracle is of the form:*

$$\mathcal{H} = \left\{ x \mapsto \sum_{t=1}^{T} \alpha_t h_t(x) \,\middle|\, T \in \mathbb{Z}_+, \alpha \in \Delta_T, h_1, \ldots, h_T \in \bar{\mathcal{H}} \right\},$$

for some class $\bar{\mathcal{H}}$ of deterministic multiclass classifiers $h : \mathcal{X} \to \{0, 1\}^k$. Suppose the deterministic hypothesis class $\bar{\mathcal{H}}$ has Natarajan dimension $d > 0$, and ϕ is 1-Lipschitz. Then for any $\delta \in (0, 1)$, as long as the sample size $n \geq O\left(\frac{d \log(k) + \log(1/\delta)}{\epsilon_\Omega^2}\right)$, the guarantee in Theorem 1 hold with probability at least $1 - \delta$ (over draw of $D = \{\mathbf{x}_i, y_i\}_{i=1}^n$ from $\mathbb{P}$).

The proof adapts generalization bounds from Daniely et al. [2015], and uses the fact that the predictive rate for any randomized classifier in $\mathcal{H}$ is a convex combination of rates for deterministic classifiers in $\bar{\mathcal{H}}$ (due to linearity of expectation).

G.2 PROOFS

Before presenting the proof of Theorem 1, we re-write the LPME guarantees from [Hiranandani et al., 2019b] for linear metrics in the presence of an oracle noise parameter ϵ_Ω from Definition 5.

Lemma 4 (LPME guarantees with oracle noise [Hiranandani et al., 2019b])**.** *Let the oracle Ω's metric be $\phi^{lin} = \langle \mathbf{a}, \mathbf{r} \rangle$ and its feedback noise parameter from Definition 5 be ϵ_Ω. Then, if the LPME procedure (Algorithm 2) is run using a sphere $\mathcal{S} \subset \mathcal{R}$ of radius ϱ and the binary-search tolerance ϵ, then by posing $O(k \log(1/\epsilon))$ queries it recovers coefficients $\hat{\mathbf{a}}$ with $\|\mathbf{a} - \hat{\mathbf{a}}\|_2 \leq O\left(\sqrt{k}(\epsilon + \sqrt{\epsilon_\Omega/\varrho})\right)$.*

Proof of Theorem 1. We first find the smoothness coefficient of the metric in Definition 3.

A function ϕ is said to be L-smooth if for some bounded constant L, we have:

$$\|\nabla\phi(x) - \nabla\phi(y)\|_2 \leq L\|x - y\|_2.$$

For the metric in Definition 3, we have:

$$\begin{aligned}\|\nabla\phi^{\text{quad}}(x) - \nabla\phi^{\text{quad}}(y)\|_2 &= \|\mathbf{a} + \mathbf{B}\mathbf{x} - (\mathbf{a} + \mathbf{B}\mathbf{y})\|_2 \\ &\leq \|\mathbf{B}\|_2\|x - y\|_2 \\ &\leq \|\mathbf{B}\|_F\|x - y\|_2, \\ &\leq 1 \cdot \|x - y\|_2,\end{aligned}$$

where in the last step, we have used the scale invariance condition from Definition 3, i.e., $\|\mathbf{a}\|_2^2 + \|\mathbf{B}\|_F^2 = 1$, which implies that $\|\mathbf{B}\|_F^2 = 1 - \|\mathbf{a}\|_2^2 \leq 1$. Hence, the metrics in Definition 3 are 1-smooth.

Now, we look at the error in Taylor series approximation when we approximate the metric ϕ^{quad} in Definition 3 with a linear approximation. Our metric is

$$\phi^{\text{quad}}(\mathbf{r}) = \langle \mathbf{a}, \mathbf{r}\rangle + \frac{1}{2}\mathbf{r}^T\mathbf{B}\mathbf{r}.$$

We approximate it with the first order Taylor polynomial around a point $\mathbf{z}$, which we define as follows:

$$T_1(\mathbf{r}) = \langle \mathbf{a}, \mathbf{z}\rangle + \frac{1}{2}\mathbf{z}^T\mathbf{B}\mathbf{z} + \langle \mathbf{a} + \mathbf{B}\mathbf{z}, \mathbf{r}\rangle$$

The bound on the error in this approximation is:

$$\begin{aligned}|E(\mathbf{r})| &= |\phi^{\text{quad}}(\mathbf{r}) - T_1(\mathbf{r})| \\ &= \frac{1}{2}|(\mathbf{r} - \mathbf{z})^T\Delta\phi^{\text{quad}}|_{\mathbf{c}}(\mathbf{r} - \mathbf{z})| && \text{(First-order Taylor approximation error)} \\ &= \frac{1}{2}|(\mathbf{r} - \mathbf{z})^T\mathbf{B}(\mathbf{r} - \mathbf{z})| && \text{(Hessian at any point } \mathbf{c} \text{ is the matrix } \mathbf{B}\text{)} \\ &\leq \frac{1}{2}\|\mathbf{B}\|_2\|\mathbf{r} - \mathbf{z}\|_2^2 \\ &\leq \frac{1}{2}\|\mathbf{B}\|_F\varrho^2 \\ &\leq \frac{1}{2}\varrho^2 && \text{(Due to the scale invariance condition)}\end{aligned}$$

So when the oracle is asked $\Omega(\mathbf{r}_1, \mathbf{r}_2) = \mathbb{1}[\phi^{\text{quad}}(\mathbf{r}_1) > \phi^{\text{quad}}(\mathbf{r}_2)]$, the approximation error can be treated as feedback error from the oracle with feedback noise $2 \times \frac{1}{2}\varrho^2$. Thus, the overall feedback noise by the oracle is $\epsilon_\Omega + \varrho^2$ for the purposes of using Lemma 4 later.

We first prove guarantees for the matrix $\mathbf{B}$ and then for the vector $\mathbf{a}$. We write Equation (9) in the following form assuming $d_1 = 1$ (since we normalize the coefficients at the end due to scale invariance):

$$\begin{aligned}B_{ij} = F_{ij} &= \left[\frac{f_{ij}}{f_{1j}}\left(1 + \frac{f_{j1}}{f_{11}}\right) - \frac{f_{ij}}{f_{1j}}\frac{f_{j0}}{f_{10}} - \frac{f_{i0}}{f_{10}} + \frac{f_{ij}}{f_{1j}}\frac{f_{j1}}{f_{11}}\frac{\frac{f_{21}^-}{f_{11}^-} + \frac{f_{21}}{f_{11}} - 2\frac{f_{20}}{f_{10}}}{\frac{f_{21}^-}{f_{11}^-} - \frac{f_{21}}{f_{11}}}\right]. \\ \implies \mathbf{B}[:, j] &= \mathbf{f}_j\left(\frac{1}{f_{1j}} + \frac{f_{j1}}{f_{1j}f_{11}} + \frac{f_{j0}}{f_{1j}f_{10}} + \frac{f_{j1}}{f_{1j}f_{11}}\left(\frac{\frac{f_{21}^-}{f_{11}^-} + \frac{f_{21}}{f_{11}} - 2\frac{f_{20}}{f_{10}}}{\frac{f_{21}^-}{f_{11}^-} - \frac{f_{21}}{f_{11}}}\right)\right) + \mathbf{f}_0\frac{1}{f_{10}} \\ &= c_j\mathbf{f}_j + c_0\mathbf{f}_0,\end{aligned} \tag{32}$$

where $\mathbf{B}[:, j]$ is the j-th column of the matrix $\mathbf{B}$, and the constants c_j and c_0 are well-defined due to the regularity Assumption 4. Notice that,

$$\frac{\partial\mathbf{B}[:, j]}{\partial\mathbf{f}_j} = diag(\mathbf{c}_j') \odot \mathbf{I} \quad \text{, and} \quad \frac{\partial\mathbf{B}[:, j]}{\partial\mathbf{f}_0} = diag(\mathbf{c}_0') \odot \mathbf{I},$$

where $\mathbf{c}'_j, \mathbf{c}'_0$ are vector of Lipschitz constants (bounded due to Assumption 4). This implies

$$\begin{aligned}\|\overline{\mathbf{B}}[:,j] - \hat{\mathbf{B}}[:,j]\|_2 &\leq c'_j\|\bar{\mathbf{f}}_j - \hat{\mathbf{f}}_j\|_2 + c'_0\|\bar{\mathbf{f}}_0 - \hat{\mathbf{f}}_0\|_2 \\ &\leq c'_j\sqrt{k}\left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right) + c'_0\sqrt{k}\left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right) \\ &= O\left(\sqrt{k}\left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right)\right),\end{aligned}$$

where we have used LPME guarantees from Lemma 4 under the oracle-feedback noise parameter $\epsilon_\Omega + \varrho^2$.

The above inequality provides bounds on each column of $\mathbf{B}$. Since $\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2$, we have $\max_{ij}|B_{ij} - \hat{B}_{ij}| \leq O\left(\sqrt{k}\left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right)\right)$, and consequentially, $\|\mathbf{B} - \hat{\mathbf{B}}\|_F \leq O\left(k\sqrt{k}\left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right)\right)$.

Now let us look at guarantees for $\mathbf{a}$. Since $\mathbf{a} = \mathbf{d} - \mathbf{B}\mathbf{o}$ from (4), we can write

$$\mathbf{a} = c_0\mathbf{f}_0 - \sum_{j=1}^{k} o_j\mathbf{B}[:,j],$$

where $c_0 = 1/f_{10}$. Since $\mathbf{o}$ is the rate achieved by random classifier, $o_j = 1/k\ \forall j \in [k]$, and thus we have

$$\frac{\partial \mathbf{a}}{\partial \mathbf{f}_0} = c_0\mathbf{I} \quad \text{and} \quad \frac{\partial \mathbf{a}}{\partial \mathbf{B}[:,j]} = \frac{1}{k}\mathbf{I}.$$

Thus,

$$\begin{aligned}\|\mathbf{a} - \hat{\mathbf{a}}\|_2 &\leq c'_0\sqrt{k}\left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right) + \frac{1}{k}\sum_{j=1}^{k}\sqrt{k}\left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right) \\ &= c'_0\sqrt{k}\left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right) + \frac{1}{k}\sum_{j=1}^{k}c'_j\sqrt{k}\left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right) \\ &= O\left(\sqrt{k}\left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right)\right),\end{aligned}$$

where c'_0, c'_j's are some Lipschitz constants (bounded due to Assumption 4). □

Notice the trade-off in the elicitation error that depends on the size of the sphere. As expected, when the radius of the sphere ϱ increases, the error due to approximation increases, but at the same time, error due to feedback reduces because we get better responses from the oracle. In contrast, when the radius of the sphere ϱ decreases, the error due to approximation decreases, but the error due to feedback increases.

The following corollary translates our guarantees on the elicited metric to the guarantees on the optimal rate of the elicited metric. This is useful in practice, because the optimal classifier (rate) obtained by optimizing a certain metric is often the key entity for many applications.

Corollary 2. *Let ϕ^{quad} be the original quadratic metric of the oracle and $\hat{\phi}^{\text{quad}}$ be its estimate obtained by the QPME procedure (Algorithm 1). Moreover, let $\mathbf{r}^*$ and $\hat{\mathbf{r}}^*$ be the mazximizers of ϕ^{quad} and $\hat{\phi}^{\text{quad}}$, respectively. Then, $\phi^{\text{quad}}(\mathbf{r}^*) \leq \phi^{\text{quad}}(\hat{\mathbf{r}}^*) + O\left(k^2\sqrt{k}\left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right)\right)$.*

Proof. We first show that if $|\phi^{\text{quad}}(r) - \hat{\phi}^{\text{quad}}(r)| \leq \epsilon$ for all rates r and some slack ϵ, then it follows that $\phi^{\text{quad}}(\hat{\mathbf{r}}^*) \geq \phi^{\text{quad}}(\mathbf{r}^*) - 2\epsilon$. This is because:

$$\begin{aligned}\phi^{\text{quad}}(\hat{\mathbf{r}}^*) &\geq \hat{\phi}^{\text{quad}}(\hat{\mathbf{r}}^*) - \epsilon && \left(\text{as } \hat{\phi}^{\text{quad}} \text{ approximates } \phi^{\text{quad}}\right) \\ &\geq \hat{\phi}^{\text{quad}}(\mathbf{r}^*) - \epsilon && \left(\text{as } \hat{\mathbf{r}}^* \text{ maximizes } \hat{\phi}^{\text{quad}}\right) \\ &\geq \phi^{\text{quad}}(\mathbf{r}^*) - 2\epsilon && \left(\text{as } \hat{\phi}^{\text{quad}} \text{ approximates } \phi^{\text{quad}}\right)\end{aligned} \tag{33}$$

Now, let us derive the trivial bound $|\phi^{\text{quad}}(r) - \hat{\phi}^{\text{quad}}(r)|$ for any rate $\mathbf{r}$.

$$
\begin{aligned}
|\phi^{\text{quad}}(r) - \hat{\phi}^{\text{quad}}(r)| &= |\langle \mathbf{a} - \hat{\mathbf{a}}, \mathbf{r} \rangle + \frac{1}{2}\mathbf{r}^T(\mathbf{B} - \hat{\mathbf{B}})\mathbf{r}| \\
&\leq |\langle \mathbf{a} - \hat{\mathbf{a}}, \mathbf{r} \rangle| + \frac{1}{2}|\mathbf{r}^T(\mathbf{B} - \hat{\mathbf{B}})\mathbf{r}| \\
&\leq \|\mathbf{a} - \mathbf{a}\|_2 \|\mathbf{r}\|_2 + \frac{1}{2}\|\mathbf{B} - \mathbf{B}\|_2 \|\mathbf{r}\|_2^2 \\
&\leq \|\mathbf{a} - \mathbf{a}\|_2 \sqrt{k} + \frac{1}{2}\|\mathbf{B} - \mathbf{B}\|_F k \\
&\leq O\left(k^2\sqrt{k}\left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right)\right),
\end{aligned} \tag{34}
$$

where in the fourth step, we have used the fact that the rates are bounded in $[0, 1]$; hence $\|\mathbf{r}\|_2 \leq \sqrt{k}$, and in the fifth step, we have used the guarantees from Theorem 1. Combining(33) and (34) gives us the desired result. □

Proof of Theorem 2. For the purpose of this proof, let us replace $\left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right)$ by some slack ϵ. Theorem 1 guarantees that after running the QPME procedure for $O(k^2 \log(1/\epsilon)$ queries, we have

- $\|a - \hat{a}\|_2 \leq O(\sqrt{k}\epsilon)$
- $\left\|B - \hat{B}\right\|_F \leq O(k\sqrt{k}\epsilon)$.

If we vectorize the tuple $(\mathbf{a}, \mathbf{B})$ and denote it by w, we have $\|w - \hat{w}\|_2 \leq O(k\sqrt{k}\epsilon)$, where both $\|w\|_2, \|\hat{w}\|_2 = 1$, due to the scale invariance condition from Definition 3. Note that w is $\frac{k^2+3k}{2}$-dimensional vector and defines the scale-invariant quadratic metric elicitation problem. Now, we have to count the minimum number of $\hat{w}$ that are possible such that $\|w - \hat{w}\|_2 \leq O(k\sqrt{k}\epsilon)$.

This translates to finding the covering number of a ball in $\|\cdot\|_2$ norm with radius 1, where the covering balls have radius $k\sqrt{k}\epsilon$. Let us denote the cover by $\{u_i\}_{i=1}^N$ and the ball with radius 1 as $\mathbb{B}$. We then have:

$$
\begin{aligned}
Vol(\mathbb{B}) =& \leq \sum_{i=1}^{N} Vol(k\sqrt{k}\epsilon\mathbb{B} + u_i) \\
&= NVol(k\sqrt{k}\epsilon\mathbb{B}) \\
&= (k\sqrt{k}\epsilon)^{\frac{k^2+3k}{2}-1}.
\end{aligned}
$$

Thus the number of $\hat{w}$ that are possible are at least

$$
c\left(\frac{1}{k\sqrt{k}\epsilon}\right)^{\frac{k^2+3k}{2}-1} \leq N,
$$

where c is a constant. Since each pairwise comparison provides at most one bit, at least $O(k^2)\log(\frac{1}{k\sqrt{k}\epsilon})$ bits are required to get a possible $\hat{w}$. We require $O(k^2)\log(\frac{1}{\epsilon})$ queries, which is near-optimal barring log terms. □

H EXTENSION TO HIGHER ORDER POLYNOMIALS

Our approach can be generalized to *higher-order polynomials* of rates. Consider e.g. a cubic polynomial:

$$
\phi^{\text{cubic}}(\mathbf{r}) := \sum_i a_i r_i + \frac{1}{2}\sum_{i,j} B_{ij} r_i r_j + \frac{1}{6}\sum_{i,j,l} C_{ijl} r_i r_j r_l,
$$

where $\mathbf{B}$ and $\mathbf{C}$ are symmetric, and $\sum_i a_i^2 + \sum_{ij} B_{ij}^2 + \sum_{ijl} C_{ijl}^2 = 1$ (w.l.o.g., due to scale invariance). A quadratic approximation to this metric around a point $\mathbf{z}$ is given by:

$$\sum_i a_i r_i + \frac{1}{2}\left(\sum_{i,j} B_{ij} r_i r_j + \sum_{i,j,l} C_{ijl}(r_i - z_i)(r_j - z_j) z_l\right) + c,$$

where c is a constant not affecting the oracle responses. We can estimate the parameters of this approximation by applying the QPME procedure from Algorithm 1 with the metric centered at an appropriate point, and its queries restricted to a small neighborhood around $\mathbf{z}$. Running QPME once using a sphere around the point $\mathbf{z}_l = \mathbf{o} + (\varrho - \varrho')\boldsymbol{\alpha}_l$, where $\varrho' < \varrho$ will elicit one face of the tensor $\mathbf{C}_{[:,:,l]}$ upto a scaling factor. Thus, it will require us to run the QPME procedure k times around the basis points $\mathbf{z}_l = \mathbf{o} + (\varrho - \varrho')\boldsymbol{\alpha}_l \;\; \forall l \in [k]$. Since we elicit scale-invariant quadratic approximation, we would need additional run of QPME procedure around the point $\mathcal{S}_{-\mathbf{z}_1}$ to elicit all the coefficients. Thus, we can recover the metric $\hat{\phi}^{\text{cubic}} = (\hat{\mathbf{a}}, \hat{\mathbf{B}}, \hat{\mathbf{C}})$ with as many queries as the number of unknowns, i.e, $\tilde{O}(k^3)$ in the cubic case.

For a d-th order polynomial, one can recursively apply this procedure to estimate $(d-1)$-th order approximations at multiple points, and similarly derive the polynomial coefficients from the estimated local approximations.

I EXTENDED EXPERIMENTS

The source code is provided along with the supplementary material. The experiments in this paper were conducted on a machine with the following configuration: *2.6 GHz 6 code Intel i7 processor with 16GB RAM.*

I.1 MORE DETAILS ON SIMULATED EXPERIMENTS ON QUADRATIC METRIC ELICITATION

Number of queries. First, we look at the number of queries that were actually required to elicit quadratic and fair (quadratic) metrics in Section 6. Recall that the QPME procedure (Algorithm 1) requires running the LPME subrotuine (Algorithm 2) $k+2$ times. As discussed in Appendix A, each run of LPME requires at most $3 \times 3 \times k \log(\pi/2\epsilon)$ queries. So, the maximum number of queries required in eliciting a quadratic metric is $(k+2) \times 3 \times 3 \times k \times 8$ for a binary search tolerance $\epsilon = 10^{-2}$, where we vary $k \in \{2, 3, 4, 5\}$ in the experiments.

However, note that, the elicitation error shown in Figure 4 is averaged over 100 simulated oracles, each one with its own simulated quadratic metric. Due to the nature of the binary search involved in the LPME subroutine (see Algorithm 2 and Figure 5), not every reduction of the search interval requires three queries. Many times the interval can be shrunk in less than three queries. The actual number of queries may vary across the oracles. The number of queries averaged over the 100 oracles corresponding to experiments in Section 6 is shown in Table 1.

Theorem 2 shows that our query complexity (which is linear in the number of unknowns) matches the lower bound for the problem, which means that it is theoretically impossible to obtain a better complexity order for our problem setup. In practice, it can be considerably reduced by making reasonable assumptions on the metric. For example, when the number of classes is large, one may just cluster the classes from error perspective. For example, one may assume same error costs for similar classes. This will reduce the number of unknowns to $O(c^2)$, where $c << k$ is the number of cluster of classes.

Comparison to a baseline. In Figures 4(a)–4(b), we show box plots of the ℓ_2 (Frobenius) norm between the true and elicited linear (quadratic) coefficients. We generally find that QPME is able to elicit metrics close to the true ones.

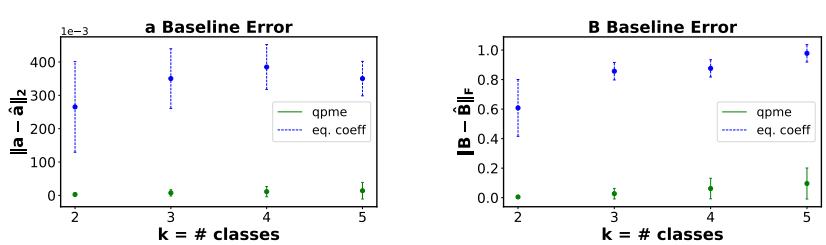


Figure 6: Elicitation error in comparison to a baseline which assigns equal coefficients.

Number of queries for QPME	
k	
2	265.43
3	669.29
4	1205.91
5	1879.74

Number of queries for Fair-QPME				
k \ m	2	3	4	5
2	332.10	867.65	1663.73	2738.59
3	796.37	2127.44	4094.31	6734.15
4	1398.14	3808.67	7363.82	12180.84
5	2130.92	5887.99	11454.18	18999.71

Table 1: Number of queries required for eliciting regular quadratic metrics (Def. 3) and fairness quadratic metrics (Def. 4) in Section 6. The number of predictive rates and sensitive groups are denoted by k and m, respectively. Recall that a quadratic metric has $O(k^2)$ unknowns. We see that the number of queries is of order $O(k^2)$ for the quadratic metric in rates, and additionally, $O(m^2k^2)$ for the fair (quadratic) metrics. Theorem 2 shows that one cannot improve on this query complexity. Nonetheless, one may make more structural assumptions on the metric to bring down the number of queries in practice.

To reinforce this point, we also compare the elicitation error of the QPME procedure and the elicitation error of a baseline which assigns equal coefficients to $\mathbf{a}$ and $\mathbf{B}$ in Figure 6. We see that the elicitation error of the baseline is order of magnitude higher than the elicitation error of the QPME procedure. This holds for varying k showing that the QPME procedure is able to elicit oracle's multiclass quadratic metrics very well.

Effect of Assumption 4. The larger standard deviation for $k = 5$ in Figure 4 is due to Assumption 4 failing to hold with sufficiently large constants $c_0, c_{-1}, c_1 \ldots, c_q$ in a small number of trials and the resulting estimates not being as accurate. We now analyze in greater detail the effect of this regularity assumption in eliciting quadratic metrics and understand how the lower bounding constants impact the elicitation error. Assumption 4 effectively ensures that the ratios computed in (9) are well-defined. To this end, we generate two sets of 100 quadratic metrics. One set is generated following Assumption 4 with one coordinate in the gradient being greater than 10^{-2}, and the other is generated randomly without any regularity condition. For both sets, we run QPME and elicit the corresponding metrics.

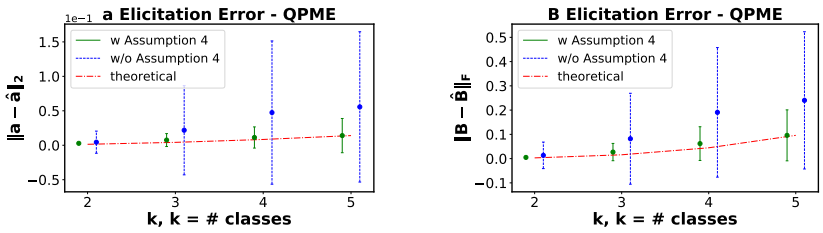


Figure 7: Elicitation error for metrics following Assumption 4 vs elicitation error for completely random metrics.

In Figure 7, we see that the elicitation error is much higher when the regularity Assumption 4 is not followed, owing to the fact that the ratio computation in (9) is more susceptible to errors when gradient coordinates approach zero in some cases of randomly generated metrics. The dash-dotted curve (in red color) shows the trajectory of the theoretical bounds with increasing k (within a constant factor). In Figure 7, we see that the mean of ℓ_2 (analogously, Frobenius) norm better follow the theoretical bound trajectory in the case when regularity Assumption 4 is followed by the metrics.

We next analyze the ratio of estimated fractions to the true fractions used in (9) over 1000 simulated runs. Ideally, this ratio should be 1, but as we see in Figure 8, these estimated ratios can be off by a significant amount for a few trials when the metrics are generated randomly. The estimated ratios, however, are more stable under Assumption 4. Since we multiply fractions in (9), even then we may observe the compounding effect of fraction estimation errors in the final estimates. Hence, we see for $k = 5$ in Figure 4(a)-4(b), the standard deviation is high due to few trials where the lower bound of 10^{-2} on the constants in Assumption 4 may not be enough. However, majority of the trials as shown in Figure 4(a)-4(b) and Figure 6 incur low elicitation error.

I.2 RANKING OF REAL-WORLD CLASSIFIERS

Performance metrics provide quantifiable scores to classifiers. This score is then often used to rank classifiers and select the best set of classifiers in practice. In this section, we discuss the benefits of elicited metrics in comparison to some default

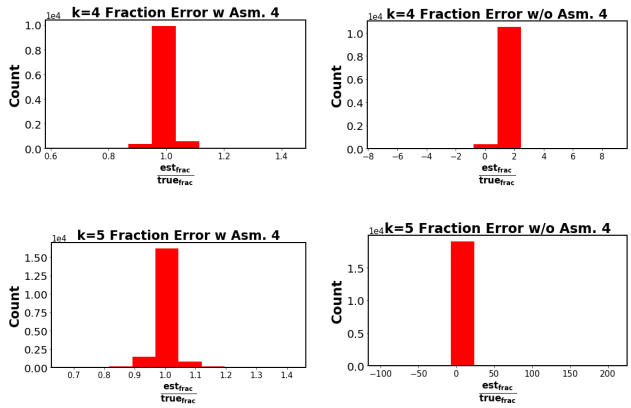


Figure 8: Ratio of estimated to true fractions over 1000 simulated runs with and without Assumption 4.

Table 2: Dataset statistics

Dataset	k	#samples	#features
default	2	30000	33
adult	2	43156	74
sensIT Vehicle	3	98528	50
covtype	7	581012	54

metrics while ranking real-world classifiers.

Ranking in case of quadratic metrics: For this experiment, we work with four real world datasets with varying number of classes $k \in \{2, 3, 7\}$. See Table 2 for details of the datasets. We use 60% of each dataset to train classifiers. The rest of the data is used to compute (testing) predictive rates. For each dataset, we create a pool of 80 classifiers by tweaking hyper-parameters in some famous machine learning models that are routinely used in practice. Specifically, we create 20 classifiers each from logistic regression models [Kleinbaum et al., 2002], multi-layer perceptron models [Pal and Mitra, 1992], LightGBM models [Ke et al., 2017], and support vector machines [Joachims, 1999]. We compare ranking of these 80 classifiers provided by competing baseline metrics with respect to the ground truth ranking, which is provided by the oracle's true metric.

We generate a random quadratic metric ϕ^{quad} following Definition 3. We treat the true ϕ^{quad} as oracle's metric. It provides us the ground truth ranking of the classifiers in the pool. We then use our proposed procedure QPME (Algorithm 1) to recover the oracle's metric. For comparison in ranking of real-world classifiers, we choose two linear metrics that are routinely employed by practitioners as baselines. The first is accuracy $\phi^{acc} = 1/\sqrt{k}\langle \mathbf{1}, \mathbf{r}\rangle$, and the second is weighted accuracy, where we just use the linear part $\langle \mathbf{a}, \mathbf{r}\rangle$ of the oracle's true quadratic metric $\langle \mathbf{a}, \mathbf{r}\rangle + \frac{1}{2}\mathbf{r}^T\mathbf{B}\mathbf{r}$. We repeat this experiment over 100 trials.

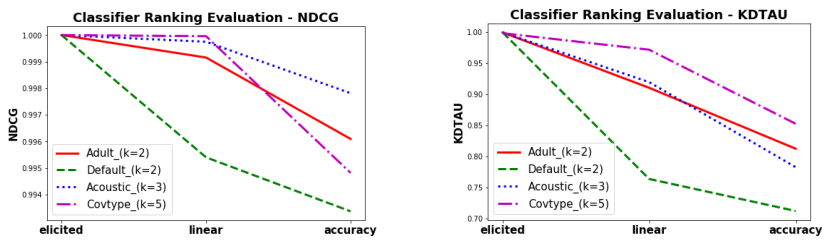


Figure 9: Performance of competing metrics while ranking real-world classifiers. 'elicited' is the metric elicited by QPME, 'linear' is the metric that comprises only the linear part of the oracle's true quadratic metric, and 'accuracy' is the linear metric which weigh all classification errors equally (often used in practice).

We report NDCG (with exponential gain) [Valizadegan et al., 2009] and Kendall-tau coefficient [Shieh, 1998] averaged over

the 100 trials in Figure 9. We observe consistently for all the datasets that the elicited metrics using the QPME procedure achieve the highest possible NDCG and Kendall-tau coefficient of 1. As we saw in Section 5, QPME may incur elicitation error, and thus the elicited metrics may not be very accurate; however, Figure 9 shows that the elicited metrics may still achieve near-optimal ranking results. This implies that when given a set of classifiers, ranking based on elicited metric scores align most closely to true ranking in comparison to ranking based on default metric scores. Consequentially, the elicited metrics may allow us to select or discard classifiers for a given task. This is advantageous in practice. For the *covtype* dataset, we see that the *linear* metric also achieves high NDCG values, so perhaps ranking at the top is quite accurate; however Kendall-tau coefficient is low suggesting that the overall ranking of classifiers is poor. We also observe that, in general, the weighted version (*linear* metric) is better than *accuracy* while ranking classifiers.

Ranking in case of fair (quadratic) metrics: With regards to fairness, we performed a similar experiment as above for comparing fair-classifiers' ranking on Adult and Default datasets with gender as the protected group. There are two genders provided in the datasets, i.e., $m = 2$. We simulate fairness metrics as given in Definition 4 that gives ground-truth ranking of classifiers and evaluate the ranking by the elicited (fair-quadratic) metric using the procedure described in Section 4 (also depicted in Figure 3). In Table 3, we show the NDCG and KD-Tau values for our method and for three baselines: (a) 'Linear with no fairness', which is the metric that comprises only the linear part of the oracle's true quadratic fair metric from Definition 4 without the fairness violation, (b) 'Accuracy with eq. odds' is the metric which weigh all classification errors and fairness violations equally, and (c) Fair Performance Metric Elicitation (FPME) procedure from [Hiranandani et al., 2020].[2] We again see that the ranking by the metric elicited using the proposed fair-QPME procedure (Section 4) is closest to the ground-truth ranking. The metric elicited by FPME [Hiranandani et al., 2020] ranks classifiers better than 'Linear with no fairness' and 'Accuracy with equalized odds'; however, it is beaten by the proposed fair-QPME procedure.

Dataset $\rightarrow$	**Adult**		**Default**	
Method $\downarrow$, **Ranking Measures** $\rightarrow$	**NDCG**	**KD-TAU**	**NDCG**	**KD-TAU**
Linear with no fairness	0.9875	0.5918	0.9994	0.9057
Accuracy with equalized odds	0.9857	0.3763	0.9889	0.4953
Elicited via FPME [Hiranandani et al., 2020]	0.9989	0.9611	0.9974	0.9650
Elicited via Fair-QPME (Proposed)	**1.0000**	**0.9972**	**1.0000**	**0.9981**

Table 3: Performance of competing metrics while ranking real-world classifiers for fairness. 'Linear with no fairness' is the metric that comprises only the linear part of the oracle's true quadratic fair metric from Definition 4 without the fairness violation, 'Accuracy with eq. odds' is the metric which weigh all classification errors and fairness violations equally (often used in practice), 'Elicited via FPME [Hiranandani et al., 2020]' is the metric elicited using the procedure from Hiranandani et al. [2020], 'Elicited via Fair-QPME' is the metric elicited by the proposed (quadratic) fairness metric elicitation procedure from Section 4 (also depicted in Figure 3),

Ranking in case of added structural assumptions on the metrics: Lastly, we discuss an experiment where we show how one may make structural assumptions on the metric when the *actual* number of unknowns is large and still get comparable results in practical settings. For this experiment, we assume that the oracle's true metric is quadratic in general rate entries as explained in Appendix E. Thus, the number of unknowns is $O(q^2)$, where $q = k^2 - k$ and is the number of off-diagonal entries of the rate matrix. We can apply the QPME procedure as it is and elicit a quadratic metric in general rates with $O(q^2)$ queries (see Appendix E), since there are $O(q^2)$ unknowns.

Note that, even if the oracle's original metric is a quadratic metric in off-diagonal entries, as a heuristic, we could still use our procedure to elicit a quadratic metric in diagonal rate entries. Moreover, we can use LPME procedure (Appendix A), too, to elicit a linear metric in off-diagonals and diagonal rate entries depending on the assumption we make on the metric.

Thus, for the ranking based experiments explained in Figure 9, we additionally ran (a) linear elicitation with diagonal rates, (b) linear elicitation with general rates, and (c) quadratic elicitation with diagonal rates, and compare their ranking with the elicited quadratic metric in general rates. As seen in Table 4, the quadratic approximation in the diagonal rates performs significantly better than eliciting a linear approximation in the general rates, while requiring the same query complexity ($\tilde{O}(k^2)$), and is close to the elicited quadratic metric in general rates, which require ($\tilde{O}(k^4)$) queries. Hence, one can make structural assumptions on the metric to reduce the query complexity and still get comparable results in practice.

[2] While FPME [Hiranandani et al., 2020] does not elicit a quadratic metric, one can still compare the elicited metrics based on how they rank candidate classifiers on real-world data.

Dataset →	Adult dataset		Default dataset	
Elicited Metric↓, Rank- Measure→	NDCG	KD-TAU	NDCG	KD-TAU
Linear-diagonal ($\tilde{O}(k)$)	0.9783	0.6053	0.9790	0.4536
Linear-general ($\tilde{O}(k^2)$)	0.9908	0.7713	0.9863	0.6216
Quadratic-diagonal ($\tilde{O}(k^2)$)	0.9968	0.9611	1.0000	0.9979
Quadratic-general ($\tilde{O}(k^4)$)	**1.0000**	**0.9986**	**1.0000**	**0.9979**

Table 4: Performance on ranking real-world classifiers when the oracle's true metric is quadratic in general rate entries: the elicited *quadratic metric in diagonal entries perform better than elicited linear metric* in general rate entries (while requiring same no. of queries), and close to the elicited quadratic metric in general rates.

J EXTENDED RELATED WORK

We discuss how the area of metric elicitation, in general, and our quadratic elicitation proposal, in particular, differ from three related fields: (i) preference elicitation in the decision theory literature, (ii) inverse reinforcement learning, (iii) ranking from pairwise comparisons using choice models.

J.1 PREFERENCE ELICITATION LITERATURE

Our work is related to decision-theoretic *preference elicitation*, however, with the following key differences. We focus on estimating the utility function (metric) explicitly, whereas prior work such as [Boutilier et al., 2006, Benabbou et al., 2017] seek to find the optimal decision via minimizing the max-regret over a set of utilities. Studies that directly learn the utility [Perny et al., 2016] do not provide query complexity guarantees for pairwise comparisons. Formulations that consider a finite set of alternatives [Boutilier et al., 2006], are starkly different than ours, because the set of alternatives in our case (i.e. classifiers or rates) is infinite. Most papers focus on linear or bilinear [Perny et al., 2016] utilities except for [Braziunas, 2012] (GAI utilities) and [Benabbou et al., 2017] (Choquet integral); whereas, we focus on quadratic metrics which are useful for classification tasks, especially, fairness.

J.2 INVERSE REINFORCEMENT LEARNING (RL)

The idea of learning a reward/cost function in the inverse RL problems is conceptually similar to metric elicitation. However, there are many key differences. Studies such as [Ng et al., 2000, Wu et al., 2020, Abbeel and Ng, 2004] try to learn a linear reward function either by knowing the optimal policy or expert demonstrations. Not only is the type of feedback in these studies different from the pairwise feedback we handle, but the studies are focused on linear rewards. In contrast, our goal is to use pairwise feedback to elicit quadratic metrics which are important for classification problems, especially, fairness. Indeed nonlinear reward estimation in inverse RL problems has been tackled before [Levine et al., 2011, Fu et al., 2017], but these are passive learning approaches and do not come with query complexity guarantees like we do. Because of the use of a complex function class, these methods are not easy to analyze. There has been some work on actively estimating the reward function in the inverse RL problems [Lopes et al., 2009]; this work involves discretizing the feature space and using maximum entropy based ideas to elicit a distribution over rewards, which clearly uses a different set of modeling assumptions than us. A recent work [Sadigh et al., 2017] elicits reward functions through active learning, but is again tied to eliciting linear functions and provides limited theoretical guarantees; whereas, we specifically focus on quadratic elicitation with rigorous guarantees.

In summary, our work is significantly different from inverse RL methods, in that unlike them, we are tied to a particular geometry of the query space (the space of error statistics achieved by feasible classifiers), and elicit quadratic (or polynomial) functions from pairwise comparisons, specifically, in an active learning manner.

J.3 RANKING FROM PAIRWISE COMPARISONS

Our work is also quite different from the use of choice models such the Bradley-Terry-Luce (BTL) model for rank aggregation. (i) Firstly, choice models such as BTL are commonly used to learn a aggregate global ranking of a finite set of N items from pairwise comparisons. The underlying problem involves estimating a N-dimensional quality score vector for the items [Shah et al., 2015]. In contrast, metric elicitation estimates an oracle's classification metric, a function of a classifier's error statistics. The applications for the two problems are very different: while ranking aggregation strategies using BTL are often prescribed for aggregating user opinions on a restaurant or product, metric elicitation seeks to find the right objective to optimize for a classification task. (ii) Secondly, the noise model in BTL is stochastic and depends on the

distance between the quality of items; whereas, our noise model in Definition 5 is not stochastic and is oblivious to the distance of rates unless the rates are very close. (iii) Thirdly, while there is some work on the extended BTL model where items are represented by feature vectors [Niranjan and Rajkumar, 2017], and the goal is to learn weights on the features to complete the ranking of the items, most of the work in this area considers a passive setting, where pairwise comparisons are assumed to be iid. In contrast, our work involves actively learning nonlinear utilities with theoretical bounds on query complexity. (iv) Lastly, the closest active learning work we could find with BTL models [Mohajer et al., 2017] does not generalize to feature-dependent utilities and is proposed for finding the top-k items, which is entirely different from metric elicitation.

Other fields that are less closely related to our work include learning scoring functions for supervised label ranking problems [Fürnkranz and Hüllermeier, 2010], and the more traditional metric learning literature, where the task is to learn a distance metric that captures similarities between data points, with the goal of using it for downstream learning tasks [Kulis et al., 2013].

K PRELIMINARY USER STUDY

We are actively conducting user studies for eliciting performance metrics. In this section, we provide a peek into our future work. The goal of this preliminary study is to check workflow of the practical implementation of the metric elicitation framework with real data, and to a certain extent, support or reject the hypothesis that the implicit user preferences can be quantified using the pairwise comparison queries over confusion matrices or predictive rates. In addition, the goal includes testing certain assumptions regarding the noise in the subject's (oracle's) responses, work around with finite samples, eliciting actual performance metrics in real-life scenarios, and evaluating the quality of the recovered metric.

The following user study works with the space of confusion matrices, i.e., entries of the type $\mathbb{P}(Y = i, h = j)$ for $i, j \in [k]$, instead of the predictive rates. In the future, we plan on incorporating rates, i.e., entries of the type $\mathbb{P}(h = j|Y = i)$ for $i, j \in [k]$, in the visualizations as well. Our contributions are summarized as follows:

- We create a web UI that uses existing visualizations of confusion matrices (predictive rates) that are refined to capture preferences over pairwise comparisons.
- The UI implements the binary-search procedure from Algorithm 2 at the back end that make use of the real-time responses over confusion matrices to elicit a linear performance metric for a binary classification task.
- We perform a user study with ten subjects and elicit their linear performance metrics using the proposed web UI. We compare the quality of the recovered metric by comparing their responses to the elicited metric's responses over a set of randomly chosen pairwise comparison queries.

K.1 CHOICE OF TASK AND DATASET USED

Our choice of task is *cancer diagnosis* [Yang and Naiman, 2014] for which we use the Breast Cancer Wisconsin (Original) dataset from the UCI repository.[3] The dataset has been extensively used in the literature for binary classification, where the label 1 denotes *malignant* cancer and label 0 denotes *benign* cancer. There are 699 samples in total, wherein each sample has 9 features. The task for any classifier is to take the 9 features of a patient as input and predict whether or not the patient has cancer. We divide this data into two equally sized parts – the training and the test data. Using the training data, we learn a logistic regression model to obtain an estimate of the class-conditional probability, i.e., $\hat{\eta}(x) = \hat{\mathbb{P}}(y = 1|X)$. We then create a sphere using Algorithm 3 inside the space of confusion matrices computed on the test data .

K.2 CHOICE OF VISUALIZATION

In modern times, ensuring effective public understanding of algorithmic decisions, especially, machine learning models has become an imperative task. With this view in mind, we borrow the visualizations of confusion matrices for the binary classifications setup from Shen et al. [2020]. The authors provide a concrete step towards the above goal by redesigning confusion matrices to support non-experts in understanding the performance of machine learning models. The final visualizations that we use from Shen et al. [2020] are created over multiple iterative user-studies. The visualizations are

[3] The dataset can be downloaded from https://tinyurl.com/dn2esyvw.

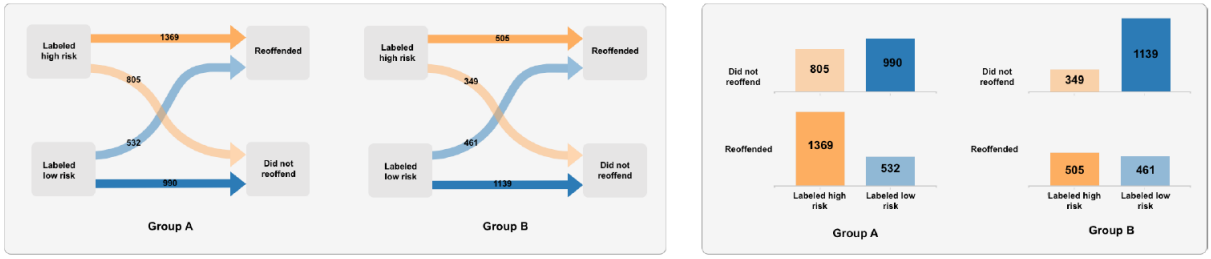


Figure 10: Flow-chart and bar-chart based visualizations for (binary classification) confusion matrices in the recidivism prediction task from Shen et al. [2020].

shown in Figure 10 in the context of a recidivism prediction task. One is the *flow-chart*, which helps users in understanding the direction of the data, and the other is the *bar chart*, which helps users in understanding the quantities involved.

However, in light of our preliminary discussions with Human-Computer Interaction (HCI) and machine learning researchers, we make/recommend the following changes in the visualization for pairwise comparison purposes in the metric elicitation framework.

1. Based on the observation that multiple visualizations of the information help in better user understanding, we choose to use both *flow-chart* and *bar-chart*, together to depict a confusion matrix.
2. We transform the data statistics so that the numbers denote out-of-100 samples.
3. We found that the total number of positive and negative labels along with total number of positive and negative predictions are very helpful in comparing two confusion matrices. Therefore, we add the total numbers in the flow-chart boxes and on axes in the bar-charts.

A sample of a pairwise comparison query with modified visualizations incorporating the points above is shown in Figure 11. We next discuss the web user interface.

K.3 USER INTERFACE

We discuss our proposed web User Interface (UI) in detail and discuss our rationale behind its components. The UI has two parts to it as explained below.

The first phase of the UI is where we actually ask subjects for pairwise preferences over confusion matrices, and implement our binary-search procedure from Algorithm 2. The subjects have to make a choice reflecting on the trade-off between false positives and false negatives. Algorithm 2 takes in real-time preferences of the subjects, generates next set of queries based on the current responses, and converge to a linear performance metric at the back end. We save this (linear) performance metric for each subject. We stop the binary-search when the search interval becomes less than or equal to 0.05 (ϵ in line 1 of Algorithm 2).

In order to evaluate the quality of the recovered metric, in the second phase, we ask the subjects fifteen pairwise comparison queries, each on a separate web page, right after the binary search algorithm has converged, and we have elicited the metric. The subjects do not know this information and are shown evaluation queries in continuation to the previous phase (i.e., the binary search). The query comprises of two randomly selected confusion matrices that lie inside the feasible region. This set of queries is used to evaluate the effectiveness of the elicited metric.

K.4 STUDY RESULTS

We compute the fraction of times our elicited metric's preferences matches with the subject's preferences on the fifteen queries, i.e.,

$$\mathcal{M} := \frac{\sum_{i=1}^{15} \mathbb{1}[\text{subject's prefer. for query } i == \text{metric's prefer. for query } i]}{15} \times 100. \tag{35}$$

We show the elicited metric for the fifteen subjects and the measure $\mathcal{M}$ values in Table 5. We see for nine out of ten subjects that more than 85% of the time our elicited metric's preferences matches with the subject's preferences on the fifteen evaluation queries. For three subjects, our metric's preference matches exactly for all the evaluation queries.

Figure 11: A sample of a pairwise comparison query from a run of the binary-search based procedure Algorithm 2

The absolute numbers for the $\mathcal{M}$ measure look good; however, how good they are is still a missing piece in this study because of the lack of a baseline. In future, we plan to devise ways to develop a baseline for the metric elicitation task and compare to that baseline on the measure $\mathcal{M}$.

Table 5: The elicited linear performance metrics for the ten subjects along with the fraction of times (in %) the elicited metric's preferences matches with the subject's preferences over the fifteen evaluation queries.

Subjects	**Linear Performance Metric**	$\mathcal{M}$
S1	0.125 TN + 0.875 TP	87
S2	0.141 TN + 0.859 TP	100
S3	0.125 TN + 0.875 TP	93
S4	0.141 TN + 0.859 TP	100
S5	0.328 TN + 0.672 TP	73
S6	0.031 TN + 0.969 TP	87
S7	0.031 TN + 0.969 TP	100
S8	0.359 TN + 0.641 TP	87
S9	0.125 TN + 0.875 TP	93
S10	0.141 TN + 0.859 TP	87