© 2021 Gaurush Hiranandani

CLASSIFICATION PERFORMANCE METRIC ELICITATION AND ITS APPLICATIONS

BY

GAURUSH HIRANANDANI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Associate Professor Oluwasanmi Koyejo, Chair
Professor Srikant Rayadurgam
Professor Paris Smaragdis
Associate Professor Shivani Agarwal, University of Pennsylvania

ABSTRACT

Given a learning problem with real-world tradeoffs, which cost function should the model be trained to optimize? This is the metric selection problem in machine learning. Despite its practical interest, there is limited formal guidance on how to select metrics for machine learning applications. This thesis outlines metric elicitation as a principled framework for selecting the performance metric that best reflects implicit user preferences. Once specified, the evaluation metric can be used to compare and train models.

In this manuscript, we formalize the problem of Metric Elicitation and devise novel strategies for eliciting classification performance metrics using pairwise preference feedback over classifiers. Specifically, we provide novel strategies for eliciting linear and linear-fractional metrics for *binary* and *multiclass* classification problems, which are then extended to a framework that elicits *group-fair* performance metrics in the presence of multiple sensitive groups. All the elicitation strategies that we discuss are robust to both finite sample and feedback noise, thus are useful in practice for real-world applications.

Using the tools and the geometric characterizations of the feasible confusion statistics space from the binary, multiclass, and multiclass-multigroup classification setups, we further provide strategies to elicit from a wider range of complex, modern multiclass metrics defined by quadratic functions of predictive rates by exploiting their local linear structure. This strategy can then be easily extended to eliciting metrics of higher order polynomials. From application perspective, we also propose to use the metric elicitation framework in optimizing complex black box metrics that is amenable to deep network training. In particular, the linear elicitation strategies can be used to elicit local-linear approximation of the black-box metrics, which are then exploited by existing iterative optimization routines. Lastly, to bring theory closer to practice, we conduct a preliminary real-user study that shows the efficacy of the metric elicitation framework in recovering the users' preferred performance metric in a binary classification setup.

“To my parents, brother, and sister-in-law for their love and support.”

ACKNOWLEDGMENTS

The only goal to pursue a Ph.D. for me was to bridge my knowledge gap. From that perspective, I could not have asked for a better *advising* and *guidance* than what my advisor, Professor Oluwasanmi Koyejo, provided. While writing this thesis, I came to believe that I have been successful in achieving my goal to a great extent. I owe every success during my Ph.D. to my esteemed advsisor, Professor Oluwasanmi Koyejo. His expertise and guidance were invaluable for my research. I will always cherish the alignment in our thinking around research work, and the kind of freedom that I had while working with you. Your support and constructive feedback on every idea that I came up with was immensely encouraging. There is always so much to learn from you, especially, the context switching between multiple projects and a very high standard for work-ethics.

I would like to thank my doctoral committee members: Professor Oluwasanmi Koyejo, Professor Srikant Rayadurgam, Professor Paris Smaragdis, and Professor Shivani Agarwal, who have always been very helpful and have given extremely thoughtful feedback on my thesis research. I am incredibly honored to be able to have them on my Ph.D. committee and feel the utmost gratitude for all their help and support.

I would like to acknowledge my colleagues from the machine-learning group. We have enjoyed sharing offices and been good friends. I would like to thank Professor Matus Telgarsky, Professor Pierre Moulin, Professor Ruoyu Sun, Professor Jiawei Han, and Professor Nan Jiang, who lectured in the outstanding courses I have taken at University of Illinois at Urbana-Champaign (UIUC). I would also like to express my gratitude to the computer science department for giving a nice and friendly environment to work with.

Without the support of my mentors, my goal of bridging the knowledge gap would not have been possible. I would like to thank Harikrishna Narasimhan, Mahdi Milani Fard, Nikhil Rao, Sumeet Katariya, Karthik Subbian, Prateek Jain, Branislav Kveton, Atanu Sinha, Shiv Kumar Saini, Sunav Choudhary, and Sumit Shekhar for their guidance and sharing of knowledge. I cannot describe in words how much I have learnt from my colleagues/co-authors at various universities and Adobe Research, where I used to work before joining the Ph.D. program. I am thankful to each and everyone of them. I would like to thank my former advisors, Professor Harish Karnick and Professor Jean-Marc Schlenker, who inspired me in the very beginning of my research career and encouraged me to pursue a doctorate degree. I would also like to thank my previous colleagues at Indian Institute of Technology, Kanpur.

Words cannot express my gratitude towards my parents, Jayshree Hiranandani and Narendra Hiranandani. They have always been there for everything! I am grateful to my parents for the sacrifices they have made in order to make me reach where I am today. I would also like to thank my brother, Dharmendra Hiranandani, who has been an inspiration and idol for me since my childhood. His guidance has been immensely helpful all throughout. The learning from our discussions over several aspects of life and career are deeply engraved inside me. They have been the force behind my achievements. I am also grateful to my sister-in-law, Ruchira Bhelekar Hiranandani, from whom I learn something each and every day regarding positive and cheerful attitude towards life.

During the writing of this thesis, I was going through ACL surgery rehabilitation. The timely deposit of this thesis could not have been possible without the support of my friends Monika Salkar, Ishita Jain, Siddhansh Agarwal, and Amber Srivastava.

Lastly, I would like to thank the Computer Science Department at UIUC for awarding me the C.L. and Jane W.-S. Liu Award, which acted as a catalyst for my research on Metric Elicitation. I would also like to thank Intel, Microsoft Azure, and Google Cloud Platform for providing computational resources to support my research.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION 1
1.1 Contributions and Thesis Organization 4

CHAPTER 2 METRIC ELICITATION 7
2.1 Preliminaries 7
2.2 Metric Elicitation: Problem Setup 8

CHAPTER 3 BINARY CLASSIFICATION PERFORMANCE METRIC ELICITATION 10
3.1 Background 11
3.2 Confusion Matrices 15
3.3 Algorithms 17
3.4 Metric Elicitation 18
3.5 Guarantees 21
3.6 Experiments 23
3.7 Related Work 25
3.8 Concluding Remarks 26

CHAPTER 4 MULTICLASS CLASSIFICATION PERFORMANCE METRIC ELICITATION 27
4.1 Preliminaries 27
4.2 Geometry and Parametrizations of the Query Spaces 31
4.3 Metric Elicitation 34
4.4 Extensions 35
4.5 Guarantees 43
4.6 Experiments 44
4.7 Discussion and Future Work 46
4.8 Related Work 46
4.9 Concluding Remarks 47

CHAPTER 5 FAIR PERFORMANCE METRIC ELICITATION 48
5.1 Background 49
5.2 Geometry of the Product Set $\mathcal{R}^{1:m}$ 53
5.3 Metric Elicitation 54
5.4 Guarantees 61
5.5 Experiments 62
5.6 Related Work 65
5.7 Concluding Remarks and Future Work 65

CHAPTER 6 QUADRATIC METRIC ELICITATION FOR FAIRNESS AND BEYOND . . . 67
6.1 Background . . . 68
6.2 Quadratic Metric Elicitation . . . 73
6.3 Eliciting Quadratic Fairness Metrics . . . 76
6.4 Guarantees . . . 80
6.5 Experiments . . . 81
6.6 Extension to Higher Order Polynomials . . . 86
6.7 Related Work . . . 87
6.8 Discussion, Limitations, and Future Work . . . 88

CHAPTER 7 OPTIMIZING BLACK-BOX METRICS THROUGH METRIC ELICITATION . . . 89
7.1 Introduction . . . 89
7.2 Problem Setup . . . 90
7.3 Example Weighting for Linear Metrics . . . 92
7.4 Plug-In Based Algorithms . . . 96
7.5 Theoretical Guarantees . . . 99
7.6 Related Work . . . 100
7.7 Experiments . . . 101
7.8 Concluding Remarks . . . 109

CHAPTER 8 PRACTICAL METRIC ELICITATION . . . 110
8.1 Dataset and Visualization Choice . . . 111
8.2 User Interface . . . 114
8.3 User Study . . . 117
8.4 Results . . . 117
8.5 Concluding Remarks . . . 122

CHAPTER 9 CONCLUSION AND FUTURE WORK . . . 131

APPENDIX A BINARY CLASSIFICATION PERFORMANCE METRIC ELICITATION . . . 134
A.1 Visualizing the Set of Confusion Matrices . . . 134
A.2 Proofs . . . 136
A.3 Extended Experiments . . . 149
A.4 Monotonically Decreasing Case . . . 152

APPENDIX B MULTICLASS CLASSIFICATION PERFORMANCE METRIC ELICITATION . . . 154
B.1 ShrinkInterval-1 and ShrinkInterval-2 Subroutines . . . 154
B.2 Proofs of Section 4.2 . . . 154
B.3 Proofs of Section 4.3 . . . 158
B.4 Proofs of Section 4.5 . . . 159
B.5 Proofs of Section 4.4 . . . 163
B.6 Extended Experiments . . . 165

APPENDIX C FAIR PERFORMANCE METRIC ELICITATION 171
C.1 Proofs and Details of Section 5.2 171
C.2 Derivations of Section 5.3 172
C.3 Proof of Section 5.4 179

APPENDIX D QUADRATIC PERFORMANCE METRIC ELICITATION 183
D.1 Geometry of the Feasible Space (Proofs of Section 6.1.2, 6.3.2) 183
D.2 Quadratic Performance Metric Elicitation Procedure 183
D.3 Fair (Quadratic) Performance Metric Elicitation Procedure 185
D.4 Elicitation Guarantee for the QPME Procedure 190

APPENDIX E OPTIMIZING BLACK-BOX METRICS THROUGH METRIC ELICITATION 195
E.1 Extension to General Linear Metrics 195
E.2 Proofs 197
E.3 Error Bound for Weight Elicitation with Fixed Probing Classifiers 209
E.4 Error Bound for FW-EG with Unknown ψ 213
E.5 Running Time of Algorithm 7.3 216
E.6 Plug-In with Coordinate-Wise Search Baseline 216
E.7 Solving Constrained Satisfaction Problem in (7.14) 217
E.8 Additional Experimental Details 219

REFERENCES 222

CHAPTER 1: INTRODUCTION

Given a class prediction problem, which performance metric should the classifier optimize? Machine learning practitioners often encounter this question in different forms. For example, natural language processing practitioners could face the question, *"What is a good summary of a given article?"* Similarly, for computer vision folks, *"What is a good caption for a given image?"* poses an identical challenge. In the field of music/audio research, the question, *"When is one piece of music similar to another?"* may get similar treatment. Medical predictions are another important application, where ignoring cost sensitive trade-offs can directly impact lives [1]. Even companies in the industry struggle to find an answer to similar questions as specialized teams of statisticians/economists are routinely hired to monitor many metrics – since optimizing the wrong metric directly translates into lost revenue [2]. Unfortunately, there is scant formal guidance within the machine learning literature for how a practitioner might choose an appropriate metric, beyond a few default choices [3, 4, 5], and even less guidance on selecting a metric that reflects the preferences of the practitioners.

To address this issue, we propose the framework of *Metric Elicitation (ME)*, where the goal is to estimate a performance metric that best reflect implicit user preferences. This framework enables a practitioner to adjust the performance metrics based on the application, context, and population at hand. The motivation is that by employing metrics that reflect a user's innate trade-offs, one can learn models that best capture the user preferences [6]. On its face, ME simply requires querying a user (oracle) to determine the quality she assigns to classifiers (learned using standard classification data); however, humans are often inaccurate when asked to provide absolute preferences [7]. Therefore, we propose gathering feedback in the form of pairwise classifier comparison queries, where the user is asked to compare two classifiers and provide an indicator of relative preference. Using such queries, ME aims to elicit the innate performance metric of the user. See Figure 1.1 for the visual intuition of the framework.

We focus on eliciting the most common performance metrics that are functions of either confusion matrix or predictive rates elements [5, 8], commonly referred as *measurements* or *classifier statistics* in this manuscript.[1] Thus, a classifier comparison query can be conceptually represented by a classifier statistics comparison query. Despite this apparent simplification, the problem remains challenging because one can only query feasible classifier statistics, i.e, classifier statistics for which there exists a classifier. To solve this problem, we introduce new characterizations of the space of feasible classifier statistics (associated

[1] Metrics depending on factors such as model complexity and interpretability are beyond the scope of this manuscript.

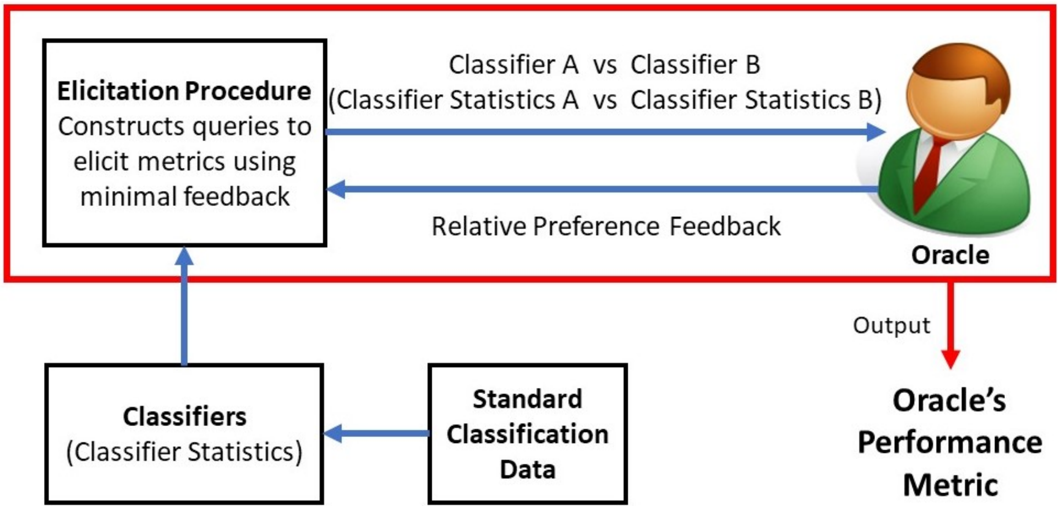

Elicitation Procedure
Constructs queries to elicit metrics using minimal feedback
Classifier A vs Classifier B
(Classifier Statistics A vs Classifier Statistics B)
Relative Preference Feedback
Oracle
Output
Oracle's Performance Metric
Classifiers
(Classifier Statistics)
Standard Classification Data

Figure 1.1: **Illustration of the Metric Elicitation Framework.** Our goal is to efficiently estimate the oracle's performance metric. We assume that the models are summarized via measurements (classifier statistics), and the metric is a function of these measurements. The elicitation procedure poses pairwise comparisons queries of the type classifier A vs classifier B (equiv. classifier statistics A vs classifier statistics B). Based on relative preference feedback from the oracle, the framework elicits the oracle's metric in as few queries as possible.

with binary, multiclass, multiclass-multigroup classification problems) enabling the design of binary-search type procedures that identify the innate performance metric of the oracle. Furthermore, all the proposed procedures remain robust, both to noise from classifier estimation and to noise in the pairwise comparison itself. Thus, our work directly results in practical algorithms. The utility of ME is illustrated via the following real life applications.

Motivating Application 1: Medical Decision-Making using Cost-Sensitive Classification. Automated medical decision-making is an important application, where ignoring cost trade-offs can directly impact lives [1]. Consider the case of cancer diagnosis and treatment support under the binary classification setting, where a doctor's unknown, innate performance metric may be approximated by a linear function of the confusion matrix elements, i.e., she has some innate reward values for True Positives and True Negatives – equivalently, costs for False Positives and False Negatives – based on known consequences of misdiagnosis, i.e, side-effects of treating a healthy patient vs. mortality rate for not treating a sick patient. Here, the doctor takes the role of the *oracle*. Our proposed approach exploits the space of confusion matrices associated with all possible classifiers that can be learned from standard classification data to determine the underlying rewards (equivalently, costs) provably using the least possible number of pairwise comparison queries posed to the doctor. Once the metric is elicited, it can be used to evaluate classifiers and/or train any future classifiers.

Motivating Application 2: Fair Machine learning. Machine learning models are increasingly applied for important decision-making tasks such as hiring and sentencing [9, 10, 11]. Yet, it is increasingly clear that automated decision-making is susceptible to bias; whereby decisions made by the algorithm are unfair to certain subgroups. To this end, several fairness metrics have been proposed – all with the goal of reducing discrimination and bias from automated decision-making [12]. One of the most difficult steps involved in practical deployment is the decision of which fairness metric to employ. This is further exacerbated by the observation that common metrics often lead to contradictory outcomes [13]. Our approach for metric elicitation can be directly used to solve the *fairness metric selection* problem. Here, perhaps groups of ethicists or other relevant decision makers take the role of the *oracle*, and group-specific predictive rates correspond to the *query space* of interest – which are easily approximated for any classifier. Metric elicitation can be used to formally quantify these intuitions – specifying the quantitative metric that is best be applied to measuring or optimizing fairness for a given machine learning task, or to quantify the tradeoff between predictive performance and fairness.

The applications of the proposed Metric Elicitation framework goes beyond just specifying user preferred performance metrics. It can also be used to learn classifiers that optimize complex performance metrics – an aspect often crucial for practical applications. Several existing optimization algorithms are iterative in nature, where in each iteration, a local-linear objective is optimized. The iterates over the optimization routine are then combined to get to the final classifier. If the form of the metric is not known, then *obtaining* the local-linear objective of the metric boils down to *eliciting linear performance metric* in a local neighborhood. Thus, the tools from the Metric Elicitation framework can be readily applied for optimizing black-box metrics. We discuss one such procedure, which optimizes black-box metrics in the presence of a *machine* oracle, that when queried for a classifier returns an absolute quality feedback for the classifier. We then briefly discuss how the proposed procedure can be extended for a *human* oracle that provides pairwise preference feedback, along with the challenges associated with it.

Lastly, we conduct a preliminary user study, where we (a) build upon existing visualizations for confusion matrices to ask for pairwise preferences, and (b) try to elicit a linear performance metric using our proposed procedure in a binary classification setup associated with cancer diagnosis. The goal of this preliminary study is to test certain assumptions, check workflow of the implementation, and provide future guidance on visualizing confusion matrices for pairwise comparisons and finally eliciting actual performance metrics in real-life scenarios.

1.1 CONTRIBUTIONS AND THESIS ORGANIZATION

We first briefly summarize the contributions from this thesis. We then dig deep into each contribution later in Chapters 2-8.

(a) **Metric elicitation framework (Chapter 2).** We formalize *Metric Elicitation (ME)* – a principled framework for determining supervised classification metrics from user feedback. For the case of pairwise feedback, we show that under certain conditions metric elicitation is equivalent to learning preferences between pairs of classifier statistics such as confusion matrices or predictive rates.

(b) **Binary classification performance metric elicitation (Chapter 3).** When the underlying metric is linear in the binary classification setup, we propose an elicitation strategy to recover the oracle's metric, whose query complexity decays logarithmically with the desired resolution. We also show that our query-complexity rates match the lower bound. We further extend the linear metric elicitation algorithm to elicit more complex yet prevalent linear-fractional binary classification performance metrics.

(c) **Multiclass classification performance metric elicitation (Chapter 4).** We extend work on binary classification setup by proposing ME strategies for the more complicated multiclass classification setting – thus significantly increasing the use cases for ME. We propose two algorithms for multiclass classification metric elicitation that use multiple binary-search subroutines that recover the oracle's linear metric. One of the proposed algorithms assumes a sparsity condition on the metric, and thus is useful when the number of classes is large. Similar to the binary case, we further provide algorithms for eliciting linear-fractional multiclass classification performance metrics.

(d) **Fair performance metric elicitation (Chapter 5).** With respect to applications to fairness, we devise a novel strategy to elicit group-fair performance metrics for multiclass classification problems with multiple sensitive groups that also includes selecting the trade-off between predictive performance and fairness violation. Our procedure exploits the *piecewise* linearity of the metric in group-specific predictive rates, uses binary-search based subroutines, and recovers the metric with linear query complexity.

(e) **Extension to quadratic metric elicitation and beyond (Chapter 6).** The previous ME strategies can only handle metrics that are linear or quasi-linear functions of classifier statistics, which can be restrictive in domains where the metrics are more complex and nuanced, e.g., [14, 15, 16]. Thus, we propose novel strategies for eliciting metrics defined by *quadratic* functions of classifier statistics, which can easily be applied to fair metric elicitation setups as well. We are thus be able to handle a more general family of metrics that can better capture a practitioner's innate preferences. We further generalize quadratic elicitation strategy to higher-order polynomial functions. The idea is to approximate a d-th order polynomial locally with $(d-1)$-th order polynomials and recursively apply our procedure to the lower-order polynomials.

(f) **Optimizing black-box metrics through metric elicitation (Chapter 7).** We consider learning to optimize a classification metric defined by a black-box function of the confusion matrix. Such black-box learning settings are ubiquitous, for example, when the learner only has query access to the metric of interest, or in noisy-label and domain adaptation applications where the learner must evaluate the metric via performance evaluation using a small validation sample. Our approach is to adaptively learn example weights on the training dataset such that the resulting weighted objective best approximates the metric on the validation sample. We use the fact that the example weights can be seen as a gradient for the metric and estimated through metric elicitation procedure, where a *machine* oracle responds with absolute quality value of a classifier on a clean validation dataset. We show how to model and estimate the example weights and use them to iteratively post-shift a pre-trained class probability estimator to construct a classifier. We also analyze the resulting procedure's statistical properties. Experiments on various label noise, domain shift, and fair classification setups confirm that our proposal compares favorably to the state-of-the-art baselines for each application.

(g) **Eliciting real-user metric preferences (Chapter 8).** Beyond technical contributions, our research raises novel questions with regards to classifier or classifier statistics visualization and interpretability for eliciting human preferences. We explore existing human-computer interface techniques for this task, including work on visualizing confusion matrices for non-expert users. We create a web user-interface and conduct a preliminary user-study in the binary classification setup in order to elicit real-users' performance metrics and devise procedures to evaluate the fidelity of the metrics that are recovered through the proposed metric elicitation framework.

All our metric elicitation procedures (contributions (a)-(e)) are shown to be robust to both finite sample and oracle feedback noise, thus are useful in practice. Our methods can be applied either by querying preferences over classifiers or classifiers statistics. Such an equivalence is crucial for practical applications [6, 17]. We provide statistical consistency guarantees of our black-box optimization algorithm (contribution (f)) that uses metric elicitation techniques in the presence of *machine* oracles. We briefly discuss how this algorithm can be extended in the presence of *human* oracles that provide pairwise feedback (including feedback from A/B tests) and the challenges associated with it. The related literature corresponding to each sub-topic is provided in the respective chapter. We draw out conclusions and future work in Chapter 9. Lastly, all the proofs are provided in the corresponding chapters' appendices.

CHAPTER 2: METRIC ELICITATION

In this section, we formally describe the problem of Metric Elicitation. We first lay out some preliminaries and standard notations corresponding to classification problems that are common to the entire manuscript.

Notation. For $k \in \mathbb{Z}_+$, we denote the index set by $[k] = \{1, 2, \cdots, k\}$ and use Δ_k to denote the $(k-1)$-dimensional simplex. We denote the inner product of vectors by $\langle \cdot, \cdot \rangle$ and the Hadamard product by $\odot$. For a matrix $\mathbf{A}$, *off-diag*$(\mathbf{A})$ returns a vector of off-diagonal elements of $\mathbf{A}$ in row-major form, and *diag*$(\mathbf{A})$ returns a vector of diagonal elements of $\mathbf{A}$. We denote the ℓ_2-norm and ℓ_∞-norm of a vector by $\| \cdot \|_2$ and $\| \cdot \|_\infty$, respectively.

2.1 PRELIMINARIES

We consider the standard k-class classification setting with $X \in \mathcal{X}$ and $Y \in [k]$ representing the input and output random variables, respectively. We assume access to a sample $\{(\mathbf{x}, y)_i\}_{i=1}^n$ of n examples generated *iid* from a distribution $\mathbb{P}(X, Y)$. We work with (randomized) classifiers

$$h : \mathcal{X} \to \Delta_k \tag{2.1}$$

that takes in a feature vector x as input and outputs its prediction in the form of a probability distribution over the k-classes. We further use

$$\mathcal{H} = \{h : \mathcal{X} \to \Delta_k\} \tag{2.2}$$

to denote the set of all classifiers.

Measurements (Classifier Statistics): We assume q measurements (classifier statistics) of each model h, with measurement functions $\{g_i : \mathcal{H} \times \mathbb{P} \to \mathbb{R}\}_{i=1}^q$. We denote the measurements (classifier statistics) of a classifier h by a vector $\mathbf{cs}(h, \mathbb{P}) = (g_1(h, \mathbb{P}), \ldots, g_q(h, \mathbb{P}))$. Examples of such statistics for a classifier include its confusion matrix $C_{ij}(h) = \mathbb{P}(Y = i, h = j)$ for $i, j \in [k]$, predictive rate matrix $R_{ij}(h) = \mathbb{P}(h = j | Y = i)$ for $i, j \in [k]$, etc.

Metrics: We consider performance metrics that are defined by a general function $\phi : [0, 1]^q \to \mathbb{R}$ of classifier statistics $\mathbf{cs}$:

$$\phi(\mathbf{cs}(h, \mathbb{P})). \tag{2.3}$$

Since the scale of the metric does not affect the learning problem [18], we allow ϕ to be bounded. Observe that for these purposes, the metric is invariant to positive multiplicative scaling and additive bias. One common example of such metrics is linear metric, which given coefficient vector $\mathbf{a} \in \mathbb{R}^q$ with $\|\mathbf{a}\|_2 = 1$ (without loss of generality, due to scale-invariance) is given by:

$$\phi^{\text{lin}} = \langle \mathbf{a}, \mathbf{cs}(h, \mathbb{P}) \rangle. \tag{2.4}$$

Feasible classifier statistics: We will restrict our attention to only those classifier statistics that are feasible, i.e., can be achieved by some classifier. This allows us to build elicitation methods that can be applied either by querying preferences over classifiers or classifiers statistics. The set of all feasible classifier statistics is given by:

$$\mathcal{CS} = \{\mathbf{cs}(h, \mathbb{P}) \,:\, h \in \mathcal{H}\}. \tag{2.5}$$

For simplicity, we will suppress the dependence on $\mathbb{P}$ and h if it is clear from the context.

2.2 METRIC ELICITATION: PROBLEM SETUP

We now describe the problem of *Metric Elicitation.* 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 underlying metric ϕ and provides answers by comparing its value on the two classifiers.

Definition 2.1 (Oracle Query)**.** Given two classifiers h_1, h_2 (equiv. to classifier statistics $\mathbf{cs}_1, \mathbf{cs}_2$ respectively), a query to the Oracle (with metric ϕ) is represented by:

$$\Gamma(h_1, h_2\,;\,\phi) = \Omega(\mathbf{cs}_1, \mathbf{cs}_2\,;\,\phi) = \mathbf{1}[\phi(\mathbf{cs}_1) > \phi(\mathbf{cs}_2)], \tag{2.6}$$

where $\Gamma : \mathcal{H} \times \mathcal{H} \to \{0, 1\}$ and $\Omega : \mathcal{CS} \times \mathcal{CS} \to \{0, 1\}$. The query asks whether h_1 is preferred to h_2 (equiv. if $\mathbf{cs}_1$ is preferred to $\mathbf{cs}_2$), 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 [19, 20] or via A/B testing [21]. For example, in an internet-based applications one may perform A/B testing by deploying two classifiers A and B with two different sub-populations of users and use their level of engagement to decide which of the two classifiers is preferred. For other applications, we may present to the user, visualizations of the measurements such as predictive rates for two different classifiers (e.g., [22, 23]), and have the user provide pairwise feedback.

Since the metrics we consider are functions of only the classifier statistics, queries comparing classifiers are the same as queries on the associated classifier statistics. So for convenience, we will have our algorithms pose queries comparing two (feasible) classifier statistics, but they can be equivalently seen as comparing two classifiers. We next formally state the ME problem.

Definition 2.2 (Metric Elicitation with Pairwise Queries (given $\mathbb{P}$))**.** Suppose that the oracle's (unknown) performance metric is ϕ. Using oracle queries of the form $\Omega(\mathbf{cs}_1, \mathbf{cs}_2\,;\,\phi)$, recover a metric $\hat{\phi}$ such that $\|\phi - \hat{\phi}\| < \kappa$ under a suitable norm $\|\cdot\|$ for sufficiently small error tolerance $\kappa > 0$.

Notice that Definition 2.2 involves true population quantities $\mathbf{cs}_1, \mathbf{cs}_2$. However, in practice, we are given only finite samples. This leads to a more practical definition of the metric elicitation problem.

Definition 2.3 (Metric Elicitation with Pairwise Queries (given $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$))**.** The same problem as stated in Definition 2.2, except that the queries are of the form $\Omega(\hat{\mathbf{cs}}_1, \hat{\mathbf{cs}}_2)$, where $\hat{\mathbf{cs}}_1, \hat{\mathbf{cs}}_2$ are the estimated classifier statistics from the given samples.

The performance of ME is evaluated by both the query complexity and the quality of the elicited metric [6, 17]. As is standard in the decision theory literature [6, 17, 24], we present our ME approach by first assuming access to population quantities such as the population classifier statistics $\mathbf{cs}(h, \mathbb{P})$ as in Definition 2.2, then examine estimation error from finite samples, i.e., with empirical rates $\hat{\mathbf{cs}}(h, \{(\mathbf{x}, y)_i\}_{i=1}^n)$ as in Definition 2.3. Lastly, in all our proposed metric elicitation strategies, we work with the following noise model:

Definition 2.4. Oracle Feedback Noise ($\epsilon_\Omega \geq 0$): The oracle may provide wrong answers whenever $|\phi(\mathbf{cs}) - \phi(\mathbf{cs}')| < \epsilon_\Omega$. Otherwise, it provides correct answers.

Simply put, if the classifier statistics $\mathbf{cs}, \mathbf{cs}'$ are close as measured by ϕ, then the oracle responses may be incorrect. We show robustness of our approaches under this noise model. We next discuss elicitation strategies for the different classification scenarios starting with the binary classification problem setup.

CHAPTER 3: BINARY CLASSIFICATION PERFORMANCE METRIC ELICITATION

In this chapter, we focus on eliciting binary classification performance metrics from pairwise feedback, where a practitioner is queried to provide relative preference between two classifiers. Here, we choose our measurement space to be the space of feasible confusion matrices associated with the classifiers for binary classification. By exploiting key geometric properties of the space of confusion matrices, we obtain provably query efficient algorithms for eliciting performance metrics. We emphasize that the notion of pairwise classifier comparison is not new and is already prevalent in the industry. An example is A/B testing [21], where the whole population of users acts as an oracle.[1] Similarly, classifier comparison by a single expert is becoming commonplace due to advances in the field of interpretable machine learning [19, 20].

In this first edition of metric elicitation strategies, we focus on the most common performance metrics which are functions of the confusion matrix [5, 8, 18], particularly, linear and ratio-of-linear functions. This includes almost all modern metrics such as accuracy, F_β-Measure, Jaccard Similarity Coefficient [5], etc. By construction, pairwise classifier comparisons may be conceptually represented by their associated pairwise confusion matrix comparisons. Despite this apparent simplification, the problem remains challenging because one can only query feasible confusion matrices, i.e. confusion matrices for which there exists a classifier. As we show, our characterization of the space of confusion matrices enables the design of efficient binary-search type procedures that identify the innate performance metric of the oracle. While classifier (confusion matrix) comparisons may introduce additional noise, our approach remains robust, both to noise from classifier (confusion matrix) estimation, and to noise in the comparison itself. Thus, our work directly results in a practical algorithm.

Example: Consider the case of cancer diagnosis, where a doctor's unknown, innate performance metric is a linear function of the confusion matrix, i.e., she has some innate reward values for True Positives and True Negatives – equivalently (equiv.), costs for False Positives and False Negatives – based on known consequences of misdiagnosis. Here, the doctor takes the role of the oracle. Our proposed approach exploit the space of confusion matrices associated with all possible classifiers that can be learned from standard classification data and determine the underlying rewards (equiv., costs) provably using the least possible number

[1] In A/B testing, sub-populations of users are shown classifier A vs. classifier B, and their responses determine the overall preference. Interestingly, while each person is shown a sample output from one of the classifiers, the entire user population acts as the oracle for comparing classifiers.

of pairwise comparison queries posed to the doctor.

Our contributions in this chapter are summarized as follows:

- When the underlying metric is linear, we propose a binary search algorithm that can recover the metric with query complexity that decays logarithmically with the desired resolution. We further show that our query-complexity rates match the lower bound.
- We extend the elicitation algorithm to more complex linear-fractional performance metrics.
- We prove robustness of the proposed approach under feedback and classifier estimation noise.

All the proofs in this chapter are provided in Appendix A.

3.1 BACKGROUND

Let $X \in \mathcal{X}$ and $Y \in \{0, 1\}$ represent the input and output random variables respectively (0 = negative class, 1 = positive class). We assume a dataset of size n, $\{(x_i, y_i)\}_{i=1}^n$, generated *iid* from a data generating distribution $\mathbb{P} \overset{\text{iid}}{\sim} (X, Y)$. Let f_X be the marginal distribution for $\mathcal{X}$. Let $\eta(x) = \mathbb{P}(Y = 1|X = x)$ and $\zeta = \mathbb{P}(Y = 1)$ represent the conditional and the unconditional probability of the positive class, respectively. Note that the earlier term is a function of the input x; whereas, the latter is a constant. We denote a classifier by h, and let $\mathcal{H} = \{h : \mathcal{X} \to [0, 1]\}$ be the set of all classifiers. A confusion matrix for a classifier h is denoted by $C(h, \mathbb{P}) \in \mathbb{R}^{2\times 2}$, comprising true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) and is given by:

$$\begin{aligned}
C_{11} &= TP(h, \mathbb{P}) = \mathbb{P}(Y = 1, h = 1), \\
C_{01} &= FP(h, \mathbb{P}) = \mathbb{P}(Y = 0, h = 1), \\
C_{10} &= FN(h, \mathbb{P}) = \mathbb{P}(Y = 1, h = 0), \\
C_{00} &= TN(h, \mathbb{P}) = \mathbb{P}(Y = 0, h = 0). \qquad (3.1)
\end{aligned}$$

Clearly, $\sum_{i,j} C_{ij} = 1$. We denote the set of all confusion matrices by $\mathcal{C} = \{C(h, \mathbb{P}) : h \in \mathcal{H}\}$. Under the population law $\mathbb{P}$, the components of the confusion matrix can be further decomposed as:

$$FN(h, \mathbb{P}) = \zeta - TP(h, \mathbb{P}) \quad \text{and} \quad FP(h, \mathbb{P}) = 1 - \zeta - TN(h, \mathbb{P}). \qquad (3.2)$$

This decomposition reduces the four dimensional space to two dimensional space. Therefore, the set of confusion matrices can be defined as

$$\mathcal{C} = \{(TP(h, \mathbb{P}), TN(h, \mathbb{P})) : h \in \mathcal{H}\}. \tag{3.3}$$

For clarity, we will suppress the dependence on $\mathbb{P}$ in our notation. In addition, we will subsume the notation h if it is implicit from the context and denote the confusion matrix by $C = (TP, TN)$.

We represent the boundary of the set $\mathcal{C}$ by $\partial\mathcal{C}$. Any hyperplane (line) ℓ in the (tp, tn) coordinate system is given by:

$$\ell := a \cdot tp + b \cdot tn = c, \qquad \text{where } a, b, c \in \mathbb{R}. \tag{3.4}$$

Let $\phi : [0, 1]^{2\times 2} \to \mathbb{R}$ be the performance metric for a classifier h determined by its confusion matrix $C(h)$. Without loss of generality (w.l.o.g.), we assume that ϕ is a utility, so that larger values are better.

3.1.1 Types of Performance Metrics

We consider two of the most common families of binary classification metrics, namely linear and linear-fractional functions of the confusion matrix (3.1).

Definition 3.1. Linear Performance Metric (LPM): We denote this family by φ_{LPM}. Given constants (representing weights) $\{a_{11}, a_{01}, a_{10}, a_{00}\} \in \mathbb{R}^4$, we define the metric as:

$$\begin{aligned} \phi(C) &= a_{11}TP + a_{01}FP + a_{10}FN + a_{00}TN \\ &= m_{11}TP + m_{00}TN + m_0, \end{aligned} \tag{3.5}$$

where $m_{11} = (a_{11} - a_{10})$, $m_{00} = (a_{00} - a_{01})$, and $m_0 = a_{10}\zeta + a_{01}(1 - \zeta)$.

Example 3.1. Weighted Accuracy (WA) [25]:

$$WA = w_1 TP + w_2 TN, \tag{3.6}$$

where $w_1, w_2 \in [0, 1]$ (w_1, w_2 can be shifted and scaled to $[0, 1]$ without changing the learning problem [18]).

Definition 3.2. Linear-Fractional Performance Metric (LFPM): We denote this family by

φ_{LFPM}. Given constants $\{a_{11}, a_{01}, a_{10}, a_{00}, b_{11}, b_{01}, b_{10}, b_{00}\} \in \mathbb{R}^8$, we define the metric as:

$$\begin{aligned}\phi(C) &= \frac{a_{11}TP + a_{01}FP + a_{10}FN + a_{00}TN}{b_{11}TP + b_{01}FP + b_{10}FN + b_{00}TN} \\ &= \frac{p_{11}TP + p_{00}TN + p_0}{q_{11}TP + q_{00}TN + q_0},\end{aligned} \tag{3.7}$$

where $p_{11} = (a_{11} - a_{10})$, $p_{00} = (a_{00} - a_{01})$, $q_{11} = (b_{11} - b_{10})$, $q_{00} = (b_{00} - b_{01})$, $p_0 = a_{10}\zeta + a_{01}(1-\zeta)$, $q_0 = b_{10}\zeta + b_{01}(1-\zeta)$.

Example 3.2. The F_β measure and the Jaccard similarity coefficient (JAC) [5]:

$$F_\beta = \frac{TP}{\frac{TP}{1+\beta^2} - \frac{TN}{1+\beta^2} + \frac{\beta^2\zeta+1-\zeta}{1+\beta^2}},\ JAC = \frac{TP}{1-TN} \tag{3.8}$$

3.1.2 Bayes Optimal and Inverse Bayes Optimal Classifiers

Given a performance metric ϕ, the Bayes utility $\overline{\tau}$ is the optimal value of the performance metric over all classifiers, i.e.,

$$\overline{\tau} = \sup_{h\in\mathcal{H}} \phi(C(h)) = \sup_{C\in\mathcal{C}} \phi(C). \tag{3.9}$$

The Bayes classifier $\overline{h}$ (when it exists) is the classifier that optimizes the performance metric, so

$$\overline{h} = \operatorname*{argmax}_{h\in\mathcal{H}} \phi(C(h)). \tag{3.10}$$

Similarly, the Bayes confusion matrix is given by

$$\overline{C} = \operatorname*{argmax}_{C\in\mathcal{C}} \phi(C). \tag{3.11}$$

We further define the inverse Bayes utility

$$\underline{\tau} = \inf_{h\in\mathcal{H}} \phi(C(h)) = \inf_{C\in\mathcal{C}} \phi(C). \tag{3.12}$$

The inverse Bayes classifier is given by

$$\underline{h} = \operatorname*{argmin}_{h\in\mathcal{H}} \phi(C(h)). \tag{3.13}$$

Similarly, the inverse Bayes confusion matrix is given by:

$$\underline{C} = \operatorname*{argmin}_{C \in \mathcal{C}} \phi(C). \tag{3.14}$$

Notice that for $\phi \in \varphi_{LPM}$ (3.5), the Bayes classifier predicts the label which maximizes the expected utility conditioned on the instance, as discussed below.

Proposition 3.1. Let $\phi \in \varphi_{LPM}$, then

$$\bar{h}(x) = \left\{ \begin{array}{ll} \mathbf{1}[\eta(x) \geq \frac{m_{00}}{m_{11}+m_{00}}], & m_{11} + m_{00} \geq 0 \\ \mathbf{1}[\frac{m_{00}}{m_{11}+m_{00}} \geq \eta(x)], & o.w. \end{array} \right\} \tag{3.15}$$

is a Bayes optimal classifier *w.r.t* ϕ. Further, the inverse Bayes classifier is given by $\underline{h} = 1 - \bar{h}$.

3.1.3 Problem Setup

We borrow the problem setup from Chapter 2, particularly, the definitions of oracle query (Definition 2.1) and Metric Elicitation with finite samples (Definition 2.3). Since our choice of measurements is the confusion matrix entries, for ease of understanding, we re-state these definitions after replacing classifier statistics by confusion matrices for binary classification.

We first formalize *oracle query.* Recall that by the definition of confusion matrices (3.1), there exists a surjective mapping from $\mathcal{H} \to \mathcal{C}$. An oracle is queried to determine relative preference between two classifiers. However, since we only consider metrics which are functions of the confusion matrix, a comparison query over classifiers becomes equivalent to a comparison query over confusion matrices in our setting.

Definition 3.3. Oracle Query: Given two classifiers h, h' (equiv. to confusion matrices C, C' respectively), a query to the Oracle (with metric ϕ) is represented by:

$$\Gamma(h, h' \,;\, \phi) = \Omega(C, C' \,;\, \phi) = \mathbf{1}[\phi(C) > \phi(C')] =: \mathbf{1}[C \succ C'], \tag{3.16}$$

where $\Gamma : \mathcal{H} \times \mathcal{H} \to \{0, 1\}$ and $\Omega : \mathcal{C} \times \mathcal{C} \to \{0, 1\}$. The query denotes whether h is preferred to h' (equiv. to C is preferred to C') as measured according to ϕ.

We emphasize that depending on practical convenience, the oracle may be asked to compare either confusion matrices or classifiers achieving the corresponding confusion matrices, via approaches discussed in the beginning of Chapter 3. Henceforth, for simplicity of notation, we will treat any comparison query as confusion matrix comparison query. Next, we state the metric elicitation problem.

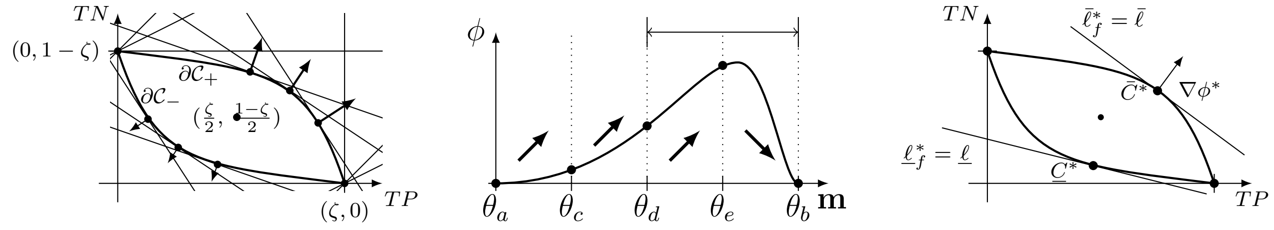


Figure 3.1: (a) Supporting hyperplanes (with normal vectors) and resulting geometry of $\mathcal{C}$; (b) Sketch of Algorithm 3.1; (c) Maximizer $\overline{C}^*$ and minimizer $\underline{C}^*$ along with the supporting hyperplanes for LFPMs.

Definition 3.4. Metric Elicitation (given $\{(x_i, y_i)\}_{i=1}^n$): Suppose that the oracle's true, unknown performance metric is ϕ. Recover a metric $\hat{\phi}$ by querying the oracle for as few pairwise comparisons of the form $\Omega(\hat{C}, \hat{C}')$, where $\hat{C}, \hat{C}'$ are the estimated confusion matrices from the samples, such that $\|\phi - \hat{\phi}\|_{--} < \kappa$ for sufficiently small $\mathbb{R} \ni \kappa > 0$ and for any suitable norm $\|\cdot\|_{--}$.

Ultimately, we want to perform ME as described in Definition 3.4. A good approach to do so is to first solve ME by assuming access to the appropriate population quantities such as the population confusion matrices $\mathbf{C}(h, \mathbb{P})$, and then consider practical implementation using estimated confusion matrices from finite data, i.e., $\mathbf{C}(h, \{(x_i, y_i)\}_{i=1}^n)$. This is a standard approach in decision theory (see e.g. [24]), where estimation error from finite samples is adjudged as a noise source and handled accordingly.

3.2 CONFUSION MATRICES

ME will require confusion matrices that are achieved by all possible classifiers, thus it is necessary to characterize the set $\mathcal{C}$ in a way which is useful for the task.

Assumption 3.1. We assume $g(t) = \mathbb{P}[\eta(X) \geq t]$ is continuous and strictly decreasing for $t \in [0, 1]$.

This is equivalent to standard assumptions [8] that the event $\eta(X) = t$ has positive density but zero probability. Note that this requires X to have no point mass.

Proposition 3.2. (Properties of $\mathcal{C}$ — Figure 3.1(a).) The set of confusion matrices $\mathcal{C}$ is convex, closed, contained in the rectangle $[0, \zeta] \times [0, 1-\zeta]$ (bounded), and 180-degree rotationally symmetric around the center-point $(\frac{\zeta}{2}, \frac{1-\zeta}{2})$. Under Assumption 3.1, $(0, 1-\zeta)$ and $(\zeta, 0)$ are the only vertices of $\mathcal{C}$, and $\mathcal{C}$ is strictly convex. Thus, any supporting hyperplane of $\mathcal{C}$ is tangent at only one point.[2]

[2] Additional visual intuition about the geometry of C (via an example) is given in Appendix A.1.

3.2.1 LPM Parametrization and Connection with Supporting Hyperplanes of $\mathcal{C}$

For an LPM ϕ (3.5), Proposition 3.2 guarantees the existence of a unique Bayes confusion matrix on the boundary $\partial\mathcal{C}$. This is because optimum for a linear function over a strictly convex set is unique and lies on the boundary [26]. Note that any linear function with the same trade-offs for *TP* and *TN*, i.e. same (m_{11}, m_{00}), is maximized at the same boundary point regardless of the bias term m_0. Thus, different LPMs can be generated by varying trade-offs $\mathbf{m} = (m_{11}, m_{00})$ such that $\|\mathbf{m}\| = 1$ and $m_0 = 0$. The condition $\|\mathbf{m}\| = 1$ does not affect the learning problem as discussed in Example 3.1. In other words, the performance metric is scale invariant. This allows us to represent the family of linear metrics φ_{LPM} by a single parameter $\theta \in [0, 2\pi]$:

$$\varphi_{LPM} = \{\mathbf{m} = (\cos\theta, \sin\theta) : \theta \in [0, 2\pi]\}. \tag{3.17}$$

Given $\mathbf{m}$ (equiv. to θ), we can recover the Bayes classifier using Proposition 3.1, and then the Bayes confusion matrix $\overline{C}_\theta = \overline{C}_\mathbf{m} = (\overline{TP}_\mathbf{m}, \overline{TN}_\mathbf{m})$ using (3.1). Under Assumption 3.1, due to strict convexity of $\mathcal{C}$, the Bayes confusion matrix $\overline{C}_\mathbf{m}$ is unique; therefore, we have that

$$\langle \mathbf{m}, C\rangle < \langle \mathbf{m}, \overline{C}_\mathbf{m}\rangle \qquad \forall\, C \in \mathcal{C}, C \neq \overline{C}_\mathbf{m}. \tag{3.18}$$

Notice the connection between the linear performance metrics and the supporting hyperplanes of the set $\mathcal{C}$ (see Figure 3.1(a)). Given $\mathbf{m}$, there exists a supporting hyperplane tangent to $\mathcal{C}$ at only $\overline{C}_\mathbf{m}$ defined as follows:

$$\bar{\ell}_\mathbf{m} := m_{11} \cdot tp + m_{00} \cdot tn = m_{11}\overline{TP}_\mathbf{m} + m_{00}\overline{TN}_\mathbf{m}. \tag{3.19}$$

Clearly, if m_{11} and m_{00} are of opposite sign (i.e., $\theta \in (\pi/2, \pi) \cup (3\pi/2, 2\pi)$), then $\bar{h}_\mathbf{m}$ is the trivial classifier predicting either 1 or 0 everywhere. In other words, if the slope of the hyperplane is positive, then it touches the set $\mathcal{C}$ either at $(\zeta, 0)$ or $(0, 1-\zeta)$. When $m_{11}, m_{00} \neq 0$ with the same sign (i.e., $\theta \in (0, \pi/2) \cup (\pi, 3\pi/2)$), then the Bayes confusion matrix is away from the two vertices. Now, we may split the boundary $\partial\mathcal{C}$ as follows:

Definition 3.5. The Bayes confusion matrices for LPMs with $m_{11}, m_{00} \geq 0$ ($\theta \in [0, \pi/2]$) form the upper boundary, denoted by $\partial\mathcal{C}_+$. The Bayes confusion matrices for LPMs with $m_{11}, m_{00} < 0$ ($\theta \in (\pi, 3\pi/2)$) form the lower boundary, denoted by $\partial\mathcal{C}_-$. From Proposition 3.1, it follows that the confusion matrices in $\partial\mathcal{C}_+$ and $\partial\mathcal{C}_-$ correspond to the classifiers of the form $\mathbf{1}[\eta(x) \geq \delta]$ and $\mathbf{1}[\delta \geq \eta(x)]$, respectively, for some $\delta \in [0, 1]$.

3.3 ALGORITHMS

In this section, we propose binary-search type algorithms, which exploit the geometry of the set $\mathcal{C}$ (Section 3.2) to find the maximizer / minimizer and the associated supporting hyperplanes for any quasiconcave / quasiconvex metrics. These algorithms are then used to elicit LPMs and LFPMs, both of which belong to both quasiconcave and quasiconvex function families.

We allow *noisy* oracles; however, for simplicity, we will first discuss algorithms and elicitation with no-noise, and then show that they are robust to the noisy feedback (Section 3.5). Moreover, as one typically prefers metrics which reward correct classification, we first discuss metrics that are monotonically increasing in both *TP* and *TN*. The monotonically decreasing case is discussed in Appendix A.4 as a natural extension.

The following lemma for any quasiconcave and quasiconvex metrics forms the basis of our proposed algorithms.

Lemma 3.1. Let $\rho^+ : [0,1] \to \partial\mathcal{C}_+$, $\rho^- : [0,1] \to \partial\mathcal{C}_-$ be continuous, bijective, parametrizations of the upper and lower boundary, respectively. Let $\phi : \mathcal{C} \to \mathbb{R}$ be a quasiconcave function, and $\psi : \mathcal{C} \to \mathbb{R}$ be a quasiconvex function, which are monotone increasing in both TP and TN. Then the composition $\phi \circ \rho^+ : [0,1] \to \mathbb{R}$ is quasiconcave (and therefore unimodal) on the interval $[0,1]$, and $\psi \circ \rho^- : [0,1] \to \mathbb{R}$ is quasiconvex (and therefore unimodal) on the interval $[0,1]$.

The unimodality of quasiconcave (quasiconvex) metrics on the upper (lower) boundary of the set $\mathcal{C}$ along with the one-dimensional parametrization of $\mathbf{m}$ using $\theta \in [0, 2\pi]$ (Section 3.2) allows us to devise binary-search-type methods to find the maximizer $\overline{C}$, the minimizer $\underline{C}$, and the first order approximation of ϕ at these points, i.e., the supporting hyperplanes at $\overline{C}$ and $\underline{C}$.

Algorithm 3.1. *Maximizing quasiconcave metrics and finding supporting hyperplanes at the optimum:* Since ϕ is monotonically increasing in both *TP* and *TN*, and $\mathcal{C}$ is convex, the maximizer must be on the upper boundary. Hence, we start with the interval $[\theta_a = 0, \theta_b = \frac{\pi}{2}]$ (Definition 3.5). We divide it into four equal parts and set slopes using (3.17) in line 4 (see Figure 3.1(b) for visual intuition). Then, we compute the Bayes classifiers using Proposition 3.1 and the associated Bayes confusion matrices in line 5. We pose four pairwise queries to the oracle in line 6. Line 7 gives the default direction to binary search in case of out-of-order responses.[3] In lines 8-12, we shrink the search interval by half based on oracle responses.

[3] Due to finite samples, $\mathcal{C}$'s boundary may have staircase-type bumps in practice. This may lead to out-of-order responses, even when the metric is unimodal *w.r.t.* θ.

Algorithm 3.1 Quasiconcave Metric Maximization

```
Input: $\epsilon > 0$ and oracle $\Omega$.
Initialize: $\theta_a = 0$, $\theta_b = \frac{\pi}{2}$.
while $|\theta_b - \theta_a| > \epsilon$ do
    Set $\theta_c = \frac{3\theta_a+\theta_b}{4}$, $\theta_d = \frac{\theta_a+\theta_b}{2}$, and $\theta_e = \frac{\theta_a+3\theta_b}{4}$. Set corresponding slopes ($\mathbf{m}$'s) using (3.17).
    Obtain $\overline{h}_{\theta_a}$,$\overline{h}_{\theta_c}$,$\overline{h}_{\theta_d}$, $\overline{h}_{\theta_e}$,$\overline{h}_{\theta_b}$ using Proposition 3.1. Compute $\overline{C}_{\theta_a}$,$\overline{C}_{\theta_c}$,$\overline{C}_{\theta_d}$,$\overline{C}_{\theta_e}$,$\overline{C}_{\theta_b}$ using (3.1).
    Query $\Omega(\overline{C}_{\theta_c}, \overline{C}_{\theta_a}), \Omega(\overline{C}_{\theta_d}, \overline{C}_{\theta_c}), \Omega(\overline{C}_{\theta_e}, \overline{C}_{\theta_d})$, and $\Omega(\overline{C}_{\theta_b}, \overline{C}_{\theta_e})$.
    If $\overline{C}_{\theta} \succ \overline{C}_{\theta'} \prec \overline{C}_{\theta''}$ for consecutive $\theta < \theta' < \theta''$, assume the default order $\overline{C}_{\theta} \prec \overline{C}_{\theta'} \prec \overline{C}_{\theta''}$.
    if $(\overline{C}_{\theta_a} \succ \overline{C}_{\theta_c})$ Set $\theta_b = \theta_d$.
    elseif $(\overline{C}_{\theta_a} \prec \overline{C}_{\theta_c} \succ \overline{C}_{\theta_d})$ Set $\theta_b = \theta_d$.
    elseif $(\overline{C}_{\theta_c} \prec \overline{C}_{\theta_d} \succ \overline{C}_{\theta_e})$ Set $\theta_a = \theta_c$, $\theta_b = \theta_e$.
    elseif $(\overline{C}_{\theta_d} \prec \overline{C}_{\theta_e} \succ \overline{C}_{\theta_b})$ Set $\theta_a = \theta_d$.
    else Set $\theta_a = \theta_d$.
end while
Output: $\overline{\mathbf{m}}, \overline{C}$, and $\bar{\ell}$, where $\overline{\mathbf{m}} = \mathbf{m}_d\ (\theta_d)$, $\overline{C} = \overline{C}_{\theta_d}$, and $\bar{\ell} := \langle \overline{\mathbf{m}}, (tp, tn) \rangle = \langle \overline{\mathbf{m}}, \overline{C} \rangle$.
```

Algorithm 3.2 Quasiconcave Metric Minimization

```
Follow Algorithm 3.1 except:
Initialize: $\theta_a = \pi$, $\theta_b = \frac{3\pi}{2}$.
Invert Responses: Replace oracle responses $C \prec C'$ with $C \succ C'$ and vice versa.
```

We stop when the search interval becomes smaller than a given $\epsilon > 0$ (tolerance). Lastly, we output the slope $\overline{\mathbf{m}}$, the Bayes confusion $\overline{C}$, and the supporting hyperplane $\bar{\ell}$ at that point.

Algorithm 3.2. *Minimizing quasiconvex metrics and finding supporting hyperplane at the optimum:* The same algorithm can be used for quasiconvex minimization with only two changes. First, we start with $\theta \in [\pi, \frac{3}{2}\pi]$, because the optimum will lie on the lower boundary $\partial\mathcal{C}_{-}$. Second, we check for $C \prec C'$ whenever Algorithm 3.1 checks for $C \succ C'$, and vice versa. Here, we output the counterparts, i.e., slope $\underline{\mathbf{m}}$, inverse Bayes Confusion matrix $\underline{C}$, and supporting hyperplane $\underline{\ell}$.

3.4 METRIC ELICITATION

In this section, we discuss how Algorithms 3.1, 3.2, and 3.3 (discussed later) are used as subroutines to elicit LPMs and LFPMs. See Figure 3.2 for a brief summary.

3.4.1 Eliciting LPMs

Suppose that the oracle's metric is $\varphi_{LPM} \ni \phi^* = \mathbf{m}^*$, where, WLOG, $\|\mathbf{m}^*\| = 1$ and $m_0^* = 0$ (Section 3.2). Application of Algorithm 3.1 to the oracle, who responds according

LPM Elicitation (True metric $\phi^* = \mathbf{m}^*$)
1. Run Algorithm 3.1 to get $\overline{C}^*$ and a hyperplane $\bar{\ell}$.
2. Set the elicited metric to be the slope of $\bar{\ell}$.

LFPM Elicitation (True metric ϕ^*)
1. Run Algorithm 3.1 to get $\overline{C}^*$, a hyperplane $\bar{\ell}$, and SoE (3.25).
2. Run Algorithm 3.2 to get $\underline{C}^*$, a hyperplane $\underline{\ell}$, and SoE (3.27).
3. Run the oracle-query independent Algorithm 3.3 to get the elicited metric, which satisfies both the SoEs.

Figure 3.2: LPM and LFPM elicitation procedures.

to $\mathbf{m}^*$, returns the maximizer and supporting hyperplane at that point. Since the true performance metric is linear, we take the elicited metric, $\hat{\mathbf{m}}$, to be the slope of the resulting supporting hyperplane.

3.4.2 Eliciting LFPMs

An LFPM is given by (3.7), where p_{11}, p_{00}, q_{11}, and q_{00} are not simultaneously zero. Also, it is bounded over $\mathcal{C}$. As scaling and shifting does not change the linear-fractional form, *w.l.o.g.*, we may take $\phi(C) \in [0,1] \, \forall C \in \mathcal{C}$ with positive numerator and denominator.

Assumption 3.2. Let $\phi \in \varphi_{LFPM}$ (3.7). We assume that $p_{11}, p_{00} \geq 0$, $p_{11} \geq q_{11}$, $p_{00} \geq q_{00}$, $p_0 = 0$, $q_0 = (p_{11} - q_{11})\zeta + (p_{00} - q_{00})(1 - \zeta)$, and $p_{11} + p_{00} = 1$.

Proposition 3.3. The conditions in Assumption 3.2 are sufficient for $\phi \in \varphi_{LFPM}$ to be bounded in $[0,1]$ and simultaneously monotonically increasing in TP and TN.

The conditions in Assumption 3.2 are reasonable as we want to elicit any unknown bounded, monotonically increasing LFPM. To no surprise, examples outlined in (3.8) and Koyejo et al. [8] satisfy these conditions. We first provide intuition for eliciting LFPMs (Figure 3.2). We obtain two hyperplanes: one at the maximizer on the upper boundary, and other at the minimizer on the lower boundary. This results in two nonlinear systems of equations (SoEs) having only one degree of freedom, but they are satisfied by the true unknown metric. Thus, the elicited metric is one where solutions to the two systems match pointwise on the confusion matrices. Formally, suppose that the oracle's metric is:

$$\phi^*(C) = \frac{p^*_{11} TP + p^*_{00} TN}{q^*_{11} TP + q^*_{00} TN + q^*_0}. \tag{3.20}$$

Let $\overline{\tau}^*$ and $\underline{\tau}^*$ be the maximum and minimum value of ϕ^* over $\mathcal{C}$, respectively, i.e.,

$$\underline{\tau}^* \leq \phi^*(C) \leq \overline{\tau}^* \; \forall \, C \in \mathcal{C}. \tag{3.21}$$

Under Assumption 3.1, we have a hyperplane

$$\overline{\ell}_f^* := (p^*_{11} - \overline{\tau}^* q^*_{11})tp + (p^*_{11} - \overline{\tau}^* q^*_{11})tn = \overline{\tau}^* q^*_0 \tag{3.22}$$

touching the set $\mathcal{C}$ only at $(\overline{TP}^*, \overline{TN}^*)$ on the upper boundary $\partial\mathcal{C}_+$. Similarly, we have a hyperplane

$$\underline{\ell}_f^* := (p^*_{11} - \underline{\tau}^* q^*_{11})tp + (p^*_{00} - \underline{\tau}^* q^*_{00})tn = \underline{\tau}^* q^*_0, \tag{3.23}$$

which touches the set $\mathcal{C}$ only at $(T\underline{P}^*, T\underline{N}^*)$ on the lower boundary $\partial\mathcal{C}_-$. To help with intuition, see Figure 3.1(c). Since LFPM is quasiconcave, Algorithm 3.1 returns a hyperplane $\overline{\ell} := \overline{m}_{11}tp + \overline{m}_{00}tn = \overline{C}_0$, where $\overline{C}_0 = \overline{m}_{11}\overline{TP}^* + \overline{m}_{00}\overline{TN}^*$. This is equivalent to $\overline{\ell}_f^*$ up to a constant multiple; therefore, the true metric is the solution to the following non-linear SoE:

$$p^*_{11} - \overline{\tau}^* q^*_{11} = \alpha\overline{m}_{11}, p^*_{00} - \overline{\tau}^* q^*_{00} = \alpha\overline{m}_{00}, \overline{\tau}^* q^*_0 = \alpha\overline{C}_0, \tag{3.24}$$

where $\alpha \geq 0$, because LHS and $\overline{m}$'s are non-negative. Additionally, we ignore the case when $\alpha = 0$, since this would imply a constant ϕ. Next, we may divide the above equations by $\alpha > 0$ on both sides so that all the coefficients $\bar{p}^*$'s and $\bar{q}^*$'s are factored by α. This does not change ϕ^*; thus, the SoE becomes:

$$p'_{11} - \overline{\tau}^* q'_{11} = \overline{m}_{11}, p'_{00} - \overline{\tau}^* q'_{00} = \overline{m}_{00}, \overline{\tau}^* q'_0 = \overline{C}_0. \tag{3.25}$$

Notice that none of the conditions in Assumption 3.2 are changed except $p'_{11} + p'_{00} = 1$. However, we may still use this condition to learn a constant α times the true metric, which does not harm the elicitation problem.

As LFPM is also quasiconvex, Algorithm 3.2 gives a hyperplane $\underline{\ell} := \underline{m}_{11}tp + \underline{m}_{00}tn = \underline{C}_0$, where $\underline{C}_0 = \underline{m}_{11}T\underline{P}^* + \underline{m}_{00}T\underline{N}^*$. This is equivalent to $\underline{\ell}_f^*$ up to a constant multiple; thus, the true metric is also the solution of the following SoE:

$$p^*_{11} - \underline{\tau}^* q^*_{11} = \gamma\underline{m}_{11}, p^*_{00} - \underline{\tau}^* q^*_{00} = \gamma\underline{m}_{00}, \underline{\tau}^* q^*_0 = \gamma\underline{C}_0, \tag{3.26}$$

where $\gamma \leq 0$ since LHS is positive, but $\underline{m}$'s are negative. Again, we may assume $\gamma < 0$. By dividing the above equations by $-\gamma$ on both sides, all the coefficients p^*'s and q^*'s are factored by $-\gamma$. This does not change ϕ^*; thus, the system of equations becomes the following:

$$p''_{11} - \underline{\tau}^* q''_{11} = \underline{m}_{11}, p''_{00} - \underline{\tau}^* q''_{00} = \underline{m}_{00}, \underline{\tau}^* q''_0 = \underline{C}_0. \tag{3.27}$$

Proposition 3.4. Under Assumption 3.2, knowing p'_{11} solves the system of equations (3.25)

Algorithm 3.3 Grid Search for Best Ratio

```
Input: $k, \Delta$.
Initialize: $\sigma_{opt} = \infty, p'_{11,opt} = 0$.
Generate $C_1, ..., C_k$ on $\partial C_+$ and $\partial C_-$ (Section 3.2).
for ($p'_{11} = 0$; $p'_{11} \leq 1$; $p'_{11} = p'_{11} + \Delta$) do
    Compute $\phi'$, $\phi''$ using Proposition 3.4. Compute array $r = [\frac{\phi'(C_1)}{\phi''(C_1)}, ..., \frac{\phi'(C_k)}{\phi''(C_k)}]$. Set $\sigma = \text{std}(r)$.
    if ($\sigma < \sigma_{opt}$) Set $\sigma_{opt} = \sigma$ and $p'_{11,opt} = p'_{11}$.
end for
Output: $p'_{11,opt}$.
```

as follows:

$$\begin{aligned} &p'_{00} = 1 - p'_{11},\, q'_0 = \overline{C}_0 \frac{P'}{Q'}, \\ &q'_{11} = (p'_{11} - \overline{m}_{11}) \frac{P'}{Q'},\, q'_{00} = (p'_{00} - \overline{m}_{00}) \frac{P'}{Q'}, \end{aligned} \tag{3.28}$$

where $P' = p'_{11}\zeta + p'_{00}(1-\zeta)$ and $Q' = P' + \overline{C}_0 - \overline{m}_{11}\zeta - \overline{m}_{00}(1-\zeta)$.

Now assume we know p'_{11}. Using Proposition 3.4, we may solve the system (3.25) and obtain a metric, say ϕ'. System (3.27) can be solved analogously, provided we know p''_{11}, to get a metric, say ϕ''. Notice that when ${}^{p^*_{11}}/_{p^*_{00}} = {}^{p'_{11}}/_{p'_{00}} = {}^{p''_{11}}/_{p''_{00}}$, then $\phi^*(C) = \phi'(C)/\alpha = -\phi''(C)/\gamma$. This means that when the true ratios of p's are known, then ϕ', ϕ'' are constant multiples of each other. So, to know the true p'_{11} (or, p''_{11}) is to search the grid $[0, 1]$ and select the one where the ratios of ϕ' and ϕ'' are constant on a number of confusion matrices. Since we can generate many confusion matrices on $\partial\mathcal{C}_+$ and $\partial\mathcal{C}_-$ (vary δ in Definition 3.5), we can estimate the ratio p'_{11} to p'_{00} using grid search based Algorithm 3.3. We may then use Proposition 3.4 for the output of Algorithm 3.3 and set the elicited metric $\hat{\phi} = \phi'$. Note that Algorithm 3.3 is independent of oracle queries and easy to implement, thus it is suitable for the purpose.

3.5 GUARANTEES

In this section, we discuss guarantees for the elicitation procedures (Section 3.4) in the presence of (a) confusion matrices' estimation noise from finite samples and (b) oracle feedback noise with the following notion that is borrowed from Definition 2.4.

Definition 3.6. Oracle Feedback Noise ($\epsilon_\Omega \geq 0$): The oracle may provide wrong answers whenever $|\phi(C) - \phi(C')| < \epsilon_\Omega$. Otherwise, it provides correct answers.

Simply put, if the confusion matrices are close as measured by ϕ, then the oracle responses can be wrong. Moving forward to the guarantees, we make two assumptions which hold in most common settings.

Assumption 3.3. Let $\{\hat{\eta}_i(x)\}_{i=1}^n$ be a sequence of estimates of $\eta(x)$ depending on the sample size. We assume that $\|\eta - \hat{\eta}_i\|_\infty \xrightarrow{P} 0$.

Assumption 3.4. For quasiconcave ϕ, recall that the Bayes classifier is of the form $h = \mathbf{1}[\eta(x) \geq \delta]$. Let $\bar{\delta}$ be the threshold that maximizes ϕ. We assume that the probability that $\eta(X)$ lies near $\bar{\delta}$ is bounded from below and above. Formally,

$$k_0\nu \leq \mathbb{P}\left[(\bar{\delta} - \eta(X)) \in [0, \nu]\right], \mathbb{P}\left[(\eta(X) - \bar{\delta}) \in [0, \nu]\right] \leq k_1\nu \tag{3.29}$$

for any $0 < \nu \leq \frac{2}{k_0}\sqrt{k_1\epsilon_\Omega}$ and some $k_1 \geq k_0 > 0$.

Assumption 3.3 is arguably natural, as most estimation is parametric, where the function classes are sufficiently well behaved. Assumption 3.4 ensures that near the optimal threshold $\bar{\delta}$, the values of $\eta(X)$ have bounded density. In other words, when X has no point mass, the slope of $\eta(X)$ where it attains the optimal threshold $\bar{\delta}$ is neither vertical nor horizontal. We start with guarantees for the algorithms in their respective tasks.

Theorem 3.1. Given $\epsilon, \epsilon_\Omega \geq 0$ and a 1-Lipschitz metric ϕ that is monotonically increasing in TP, TN. If it is quasiconcave (quasiconvex) then Algorithm 3.1 (Algorithm 3.2) finds an approximate maximizer $\overline{C}$ (minimizer $\underline{C}$). Furthemore, (i) the algorithm returns the supporting hyperplane at that point, (ii) the value of ϕ at that point is within $O(\sqrt{\epsilon_\Omega} + \epsilon)$ of the optimum, and (iii) the number of queries is $O(\log\frac{1}{\epsilon})$.

Lemma 3.2. Under our model, no algorithm can find the maximizer (minimizer) in fewer than $O(\log\frac{1}{\epsilon})$ queries.

Theorem 3.1 and Lemma 3.2, guarantee that Algorithm 3.1 (Algorithm 3.2), for a quasiconcave (quasiconvex) metric, finds a confusion matrix and a hypeplane which is close to the true maximizer (minimizer) and its associated supporting hyperplane, using just the optimal number of queries. Further, since binary search always tends towards the optimal whenever responses are correct, the algorithms necessarily terminate within a confidence interval of the true maximizer. Thus, we can take ϵ sufficiently small so that the only error that arises is due to the feedback noise ϵ_Ω. Now, we present our main result which guarantees effective LPM elicitation. Guarantees in LFPM elicitation follow naturally as discussed in the proof of Theorem 3.2 (Appendix A.2).

Table 3.1: LPM elicitation at tolerance $\epsilon = 0.02$ radians.

$\phi^* = \mathbf{m}^*$	$\hat{\phi} = \hat{\mathbf{m}}$	$\phi^* = \mathbf{m}^*$	$\hat{\phi} = \hat{\mathbf{m}}$
(0.98,0.17)	(0.99,0.17)	(-0.94,-0.34)	(-0.94,-0.34)
(0.64,0.77)	(0.64,0.77)	(-0.50,-0.87)	(-0.50,-0.87)

Theorem 3.2. Let $\varphi_{LPM} \ni \phi^* = \mathbf{m}^*$ be the true performance metric. Under Assumption 3.4, given $\epsilon > 0$, LPM elicitation (Section 3.4.1) outputs a performance metric $\hat{\phi} = \hat{\mathbf{m}}$, such that $\|\mathbf{m}^* - \hat{\mathbf{m}}\|_\infty \leq \sqrt{2}\epsilon + \frac{2}{k_0}\sqrt{2k_1\epsilon_\Omega}$.

So far, we assumed access to the confusion matrices. However, in practice, we need to estimate them using samples $\{(x_i, y_i)\}_{i=1}^n$. We now discuss robustness of the algorithms working with samples. Recall that, as a standard consequence of Chernoff-type bounds [27], sample estimates of true-positive and true-negative are consistent estimators. Therefore, with high probability, we can estimate the confusion matrix within any desired tolerance, provided we have sufficient samples. This implies that we can also estimate the ϕ values within any tolerance since LPM and LFPM are 1-Lipschitz due to (3.17) and Assumption 3.2, respectively. Thus, with high probability, the elicitation procedures gather correct oracle's preferences within feedback noise ϵ_Ω. Further, we may prove the following lemma which allow us to control the error in optimal classifiers from using the estimated $\hat{\eta}(x)$ rather than the true $\eta(x)$.

Lemma 3.3. Let h_θ and $\hat{h}_\theta$ be two classifiers estimated using η and $\hat{\eta}$, respectively. Further, let $\bar{\theta}$ be such that $h_{\bar{\theta}} = \operatorname{argmax}_\theta \phi(h_\theta)$. Then $\|C(\hat{h}_{\bar{\theta}}) - C(h_{\bar{\theta}})\|_\infty = O(\|\hat{\eta}_n - \eta\|_\infty)$.

The errors due to using $\hat{\eta}$, instead of true η may propel in the results discussed earlier, however, only in the bounded sense. This shows that our elicitation approach is robust to feedback and finite sample noise.

3.6 EXPERIMENTS

In this section, we empirically validate the theory and investigate the sensitivity due to sample estimates.[4]

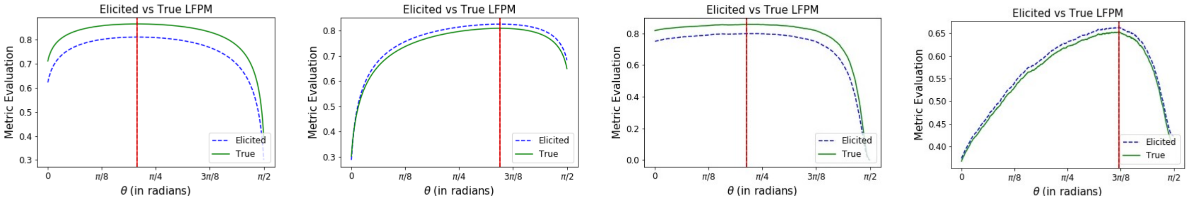
3.6.1 Synthetic Data Experiments

We assume a joint probability for $\mathcal{X} = [-1, 1]$ and $\mathcal{Y} = \{0, 1\}$ given by $f_X = \mathbb{U}[-1, 1]$ and $\eta(x) = \frac{1}{1+e^{ax}}$, where $\mathbb{U}[-1, 1]$ is the uniform distribution on $[-1, 1]$, and a is a parameter

[4] A subset of results is shown here. Please refer Appendix A.3 for extended set of results.

Table 3.2: LFPM Elicitation for synthetic distribution (Section 3.6.1) and Magic (M) dataset (Section 3.6.2). α and σ are the mean and standard deviation of $\hat{\phi}/\phi^*$ evaluated over a subset of confusion matrices used in Algorithm 3.3.

True Metric	Results on Synthetic Distribution (Section 3.6.1)			Results on Real World Dataset M (Section 3.6.2)		
$(p^*_{11}, p^*_{00}), (q^*_{11}, q^*_{00}, q^*_0)$	$(\hat{p}_{11}, \hat{p}_{00}), (\hat{q}_{11}, \hat{q}_{00}, \hat{q}_0)$	α	σ	$(\hat{p}_{11}, \hat{p}_{00}), (\hat{q}_{11}, \hat{q}_{00}, \hat{q}_0)$	α	σ
(1.00,0.00),(0.50,-0.50,0.50)	(1.00,0.00),(0.25,-0.75,0.75)	0.92	0.03	(1.00,0.00),(0.25,-0.75,0.75)	0.90	0.06
(0.20,0.80),(-0.40,-0.20,0.80)	(0.12, 0.88),(-0.43, 0.002, 0.71)	1.02	0.006	(0.19,0.81),(-0.38,-0.13,0.70)	1.02	0.004



(a) Table 3.2, line 1, col 2 (b) Table 3.2, line 2, col 2 (c) Table 3.2, line 1, col 5 (d) Table 3.2, line 2, col 5

Figure 3.3: True (solid green) and elicited (dashed blue) LFPMs for synthetic distribution and dataset M from Table 3.2. The solid red and coinciding dashed black vertical lines are *argmax* of the true and elicited metric, respectively.

controlling the degree of noise in the labels. We fix $a = 5$ in our experiments. To verify LPM elicitation, we first define a true metric ϕ^*. This specifies the query outputs in line 6 of Algorithm 3.1 (Algorithm 3.2). Then we run LPM elicitation procedure (Section 3.4.1) to check whether or not we compute the same metric. Some results are shown in Table 3.1. We elicit the true metrics even for $\epsilon = 0.02$ radians.

Next, we elicit LFPM. We define a true metric ϕ^* by $\{(p^*_{11}, p^*_{00}), (q^*_{11}, q^*_{00}, q^*_0)\}$. Then we follow the LFPM elicitation procedure (Section 3.4.2), where Algorithms 3.1 and 3.2 are run with $\epsilon = 0.05$ and Algorithm 3.3 is run with $k = 2000$ and $\Delta = 0.01$. The elicited metric $\hat{\phi}$ is denoted by $\{(\hat{p}_{11}, \hat{p}_{00}), (\hat{q}_{11}, \hat{q}_{00}, \hat{q}_0)\}$ and presented in Table 3.2 (Column 2). We also present mean (α) and standard deviation (σ) of the ratio of the elicited metric $\hat{\phi}$ to the true metric ϕ^* over a subset of confusion matrices (columns 3 and 4). For improved comparisons, Figure 3.3 shows the true and elicited metrics evaluated on selected pairs of $(TP, TN) \in \partial\mathcal{C}_+$. The metrics are plotted together after sorting the slope parameter θ. Clearly, the elicited metric is a constant multiple of the true metric. We also see that the *argmax* of the true and elicited metric coincide, thus validating Theorem 3.1.

3.6.2 Real-World Data Experiments

Now, we validate the elicitation procedures with two real-world datasets. The datasets are: (a) Breast Cancer (BC) Wisconsin Diagnostic dataset [28] containing 569 instances, and (b)

Magic (M) dataset [29] containing 19020 instances. For both the datasets, we standardize the features and split the data into two parts $\mathcal{S}_1$ and $\mathcal{S}_2$. On $\mathcal{S}_1$, we learn the estimator $\hat{\eta}$ using regularized logistic regression model. We use $\mathcal{S}_2$ for making predictions and computing sample confusion matrices.

We randomly selected twenty-eight LPMs by choosing θ^* ($\mathbf{m}^*$). We then used Algorithm 3.1 (Algortihm 3.2) with different tolerance ϵ and for different datasets and recovered the estimate $\hat{\mathbf{m}}$ using LPM elicitation. In Table A.3 of Appendix A.3, we report the proportion of the number of times when our procedure failed to recover the true $\mathbf{m}^*$. We see improved elicitation for dataset M, suggesting that ME improves with larger datasets. In particular, for dataset M, we elicit all the metrics within threshold $\epsilon = 0.11$ radians. We also observe that $\epsilon = 0.02$ is an overly tight tolerance for both the datasets leading to many failures. This is because the elicitation routine gets stuck at the closest achievable confusion matrix from finite samples, which need not be optimal within the given (small) tolerance.

Next, we evaluate LFPM elicitation using dataset M. We define the same true metrics and follow the same LFPM elicitation process as defined in Section 3.6.1. In Table 3.2 (columns 5, 6, and 7), we present the elicitation results along with mean α and standard deviation σ of the ratio of the elicited metric and the true metric. We also show the true and elicited metrics evaluated on the selected pairs of $(TP, TN) \in \partial\mathcal{C}_+$ in Figure 3.3, ordered by the parameter θ. We see that the elicited metrics are equivalent to the true metrics up to a constant.

3.7 RELATED WORK

Our work may be compared to ranking from pairwise comparisons [30]. However, we note that our results depend on novel geometric ideas on the space of confusion matrices. Thus, instead of a ranking problem, we show that ME in standard models can be reduced to just finding the maximizer (and minimizer) of an unknown function which in turn yields the true metric – resulting in low query complexity. A direct ranking approach adds unnecessary complexity to achieve the same task. Further, in contrast to our approach, most large margin ordinal regression based ranking [31] fail to control which samples are queried. There is another line of work, which actively controls the query samples for ranking, e.g., [32]. However, to our knowledge, this requires that the number of objects is finite and finite dimensional – thus cannot be directly applied to ME without significant modifications, e.g. exploiting confusion matrix properties, as we have. Learning a performance metric which correlates with human preferences has been studied before [33, 34]; however, these studies learn a regression function over some predefined features which is fundamentally different from our problem. Lastly, while [3, 4] address how one might qualitatively choose between metrics, none addresses our central contribution – a principled approach for eliciting the ideal metric from user feedback.

3.8 CONCLUDING REMARKS

We conceptualize *metric elicitation* for the binary classification setup and elicit linear and linear-fractional metrics using preference feedback over pairs of classifiers. We propose provably query efficient and robust algorithms to elicit metrics that exploit key geometric properties of the set of confusion matrices associated with the binary classification tasks.

CHAPTER 4: MULTICLASS CLASSIFICATION PERFORMANCE METRIC ELICITATION

Conceptually, Metric Elicitation (ME) is applicable to any learning setting. However, the proposed methods in the previous chapter were limited to eliciting binary classification performance metrics. This chapter extends the previous work by proposing ME strategies for the more complicated multiclass classification setting – thus significantly increasing the use cases for ME. Similar to the binary case, we consider the most common families of performance metrics which are functions of the confusion matrix [18], which is our choice of measurement space in this chapter; however, in this case, the elements of the confusion matrix summarize multiclass error statistics.

In order to perform efficient multilcass performance metric elicitation, we study novel geometric properties of the space of multiclass confusion matrices. Our analysis reveals that due to structural differences between the space of binary and multiclass confusions, we can not trivially extend the elicitation procedure used for binary to the multiclass case. Instead, we provide novel strategies for eliciting linear functions of the multiclass confusion matrix and extend elicitation to more complicated yet popular functional forms such as linear-fractional functions of the confusion matrix elements [15]. Specifically, the elicitation procedures involve binary-search type algorithms that are robust to both finite sample and oracle feedback noise. In addition, the proposed methods can be applied either by querying pairwise classifier preferences or pairwise confusion matrix preferences.

In summary, our main contributions are novel query efficient metric elicitation algorithms for multiclass classification. We first study ME for linear functions of the confusion matrix and then discuss extensions to more complicated functional forms such as the linear-fractional and arbitrary monotonic functions of the confusion matrix. Lastly, we show that the proposed procedures are robust to finite sample and feedback noise, thus are useful in practice. All the proofs in this chapter are provided in Appendix B.

Notation. Matrices and vectors are denoted by bold upper case and bold lower case letters, respectively. Recall that, given a matrix $\mathbf{A}$, *off-diag*$(\mathbf{A})$ returns a vector of off-diagonal elements of $\mathbf{A}$ in row-major form, and *diag*$(\mathbf{A})$ returns a vector of diagonal elements of $\mathbf{A}$. $\|\cdot\|_1$, $\|\cdot\|_2$, and $\|\cdot\|_\infty$ denote the ℓ_1-norm, ℓ_2-norm, and ℓ_∞-norm, respectively.

4.1 PRELIMINARIES

The standard multiclass classification setting comprises k classes with $X \in \mathcal{X}$ and $Y \in [k]$ representing the input and output random variables, respectively. We have access to a

Table 4.1: The Bayes Optimal (BO) and Restricted-Bayes Optimal (RBO) entities.

Name	Definition
BO confusion $\bar{\mathbf{c}}$ over a subset $\mathcal{S} \subseteq \mathcal{C}$	$\operatorname*{argmax}_{\mathbf{c} \in \mathcal{S} \subseteq \mathcal{C}} \phi(\mathbf{c})$
RBO classifier $\bar{h}_{k_1,k_2}$	$\operatorname*{argmax}_{h \in \mathcal{H}_{k_1,k_2}} \psi(\mathbf{d}(h))$
RBO diagonal confusion $\bar{\mathbf{d}}_{k_1,k_2}$	$\operatorname*{argmax}_{\mathbf{d} \in \mathcal{D}_{k_1,k_2}} \psi(\mathbf{d})$

dataset of size n denoted by $\{(\mathbf{x}, y)_i\}_{i=1}^n$, generated *iid* from a distribution $\mathbb{P}(X, Y)$. Let $\eta_i(\mathbf{x}) = \mathbb{P}(Y = i|X = \mathbf{x})$ and $\zeta_i = \mathbb{P}(Y = i)$ for $i \in [k]$ be the conditional and the unconditional probability of the k classes, respectively. Let $\mathcal{H} = \{h : \mathcal{X} \to \Delta_k\}$ be the set of all classifiers. A confusion matrix for a classifier h is denoted by $\mathbf{C}(h, \mathbb{P}) \in \mathbb{R}^{k\times k}$, where its elements are given by:

$$C_{ij}(h, \mathbb{P}) = \mathbb{P}(Y = i, h = j) \quad \text{for } i, j \in [k]. \tag{4.1}$$

Under the population law $\mathbb{P}$, it is useful to keep the following decomposition in mind:

$$\mathbb{P}(Y = i, h = i) = \zeta_i - \mathbb{P}(Y = i, h \neq i) \implies C_{ii}(h, \mathbb{P}) = \zeta_i - \sum_{j=1, j\neq i}^{k} C_{ij}(h, \mathbb{P}). \tag{4.2}$$

Using this decomposition, any confusion matrix is uniquely represented by its $q := (k^2 - k)$ off-diagonal elements. Hence, we will represent a confusion matrix $\mathbf{C}(h, \mathbb{P})$ by a vector $\mathbf{c}(h, \mathbb{P}) = \textit{off-diag}(\mathbf{C}(h, \mathbb{P}))$, and interchangeably refer the confusion matrix as a vector of *'off-diagonal confusions'*. The space of off-diagonal confusions is denoted by

$$\mathcal{C} = \{\mathbf{c}(h, \mathbb{P}) = \textit{off-diag}(\mathbf{C}(h, \mathbb{P})) : h \in \mathcal{H}\}. \tag{4.3}$$

For clarity, we will suppress the dependence on $\mathbb{P}$ and h if it is clear from the context.

Performance of a classifier is often determined by just the misclassification and not the type of misclassification, especially when the number of classes is large. Therefore, we will also consider metrics that only depend on correct and incorrect predictions, namely $\mathbb{P}(Y = i, h = i)$ and $\mathbb{P}(Y = i, h \neq i)$. Following the decomposition in (4.2), such metrics require only the diagonal elements of the original confusion matrices. Given a confusion matrix $\mathbf{C}$, we will denote its diagonal by $\mathbf{d} = \textit{diag}(\mathbf{C})$ and refer it as the vector of *'diagonal confusions'*. The space of diagonal confusions is represented by

$$\mathcal{D} = \{\mathbf{d} = \textit{diag}(\mathbf{C}(h)) : h \in \mathcal{H}\}. \tag{4.4}$$

Let $\phi : [0,1]^q \to \mathbb{R}$ and $\psi : [0,1]^k \to \mathbb{R}$ be the performance metrics for a classifier h determined by its corresponding off-diagonal and diagonal confusion entries $\mathbf{c}(h)$ and $\mathbf{d}(h)$, respectively. Without loss of generality (w.l.o.g.), we assume the metrics ϕ and ψ are utilities so that larger values are preferred. Furthermore, the metrics are scale invariant as global scale does not affect the learning problem [18]. For this chapter, we assume the following regularity assumption on the data distribution.

Assumption 4.1. We assume that the functions $g_{ij}(r) = \mathbb{P}\left[\frac{\eta_i(X)}{\eta_j(X)} \geq r\right] \forall\, i, j \in [k]$ are continuous and strictly decreasing for $r \in [0, \infty)$.

Intuitively, this weak assumption ensures that when the cost or reward tradeoffs for the classes change, the preferred confusions for those tradeoffs also change (and vice-versa).

4.1.1 Bayes Optimal and Restricted Bayes Optimal Confusions and Classifiers

As illustrated in Table 4.1, the Bayes Optimal (BO) confusion $\bar{\mathbf{c}}$ represents the optimal value of the off-diagonal confusions according to the metric ϕ over a subset $\mathcal{S} \subseteq \mathcal{C}$. This is analogously defined for ψ and $\mathcal{D}$. The Restricted Bayes Optimal (RBO) entities are of interest for diagonal metrics ψ, and indicate the case where classifiers are 'restricted' to predict only classes $k_1, k_2 \in [k]$. Thus $\mathcal{H}_{k_1,k_2}$ and $\mathcal{D}_{k_1,k_2}$ denote the space of classifiers which exclusively predict either k_1 or k_2 and the associated space of diagonal confusions, respectively. Note that for such restricted classifiers h, $C_{ii}(h) = d_i(h)$ evaluates to zero at every index $i \neq k_1, k_2$.

4.1.2 Performance Metrics

We first discuss elicitation for the following two major types of metrics.

Definition 4.1. Diagonal Linear Performance Metric (DLPM): We denote this family by φ_{DLPM}. Given $\mathbf{a} \in \mathbb{R}^k$ such that $\|\mathbf{a}\|_1 = 1$ (w.l.o.g., due to scale invariance), the metric is defined as:

$$\psi(\mathbf{d}) \coloneqq \langle \mathbf{a}, \mathbf{d} \rangle. \tag{4.5}$$

This is also called weighted accuracy [18] and focuses on correct classification.

Definition 4.2. Linear Performance Metric (LPM): We denote this family by φ_{LPM}. Given $\mathbf{a} \in \mathbb{R}^q$ such that $\|\mathbf{a}\|_2 = 1$ (w.l.o.g., due to scale invariance), the metric is defined as:

$$\phi(\mathbf{c}) \coloneqq \langle \mathbf{a}, \mathbf{c} \rangle. \tag{4.6}$$

Cost-sensitive linear metrics belong to φ_{LPM} [35] and focus on the types of misclassifications.

The difference of norms in the definitions is only for simplicity of exposition and chosen to best complement the underlying metric elicitation algorithm and vice-versa. Moreover, notice that the elements of diagonal confusions ($\mathbf{d}$'s) and off-diagonal confusions ($\mathbf{c}$'s) reflect correct and incorrect classification, respectively. Thus, according to standard practice, w.l.o.g., we focus on eliciting monotonically increasing DLPMs and monotonically decreasing LPMs in their respective arguments.

4.1.3 Metric Elicitation; Problem Setup

This section describes the problem of *Metric Elicitation* and the associated *oracle query*. Our definitions follow from Chapter 2, extended so the confusion elements and the performance metrics correspond to the multiclass classification setting. The following definitions hold analogously for the diagonal case by replacing $\phi, \mathbf{c}$ and $\mathcal{C}$ by $\psi, \mathbf{d}$, and $\mathcal{D}$, respectively.

Definition 4.3 (Oracle Query)**.** Given two classifiers h, h' (equivalent to off-diagonal confusions $\mathbf{c}, \mathbf{c}'$ respectively), a query to the Oracle (with metric ϕ) is represented by:

$$\Gamma(h, h'\,;\,\phi) = \Omega(\mathbf{c}, \mathbf{c}'\,;\,\phi) = \mathbf{1}[\phi(\mathbf{c}) > \phi(\mathbf{c}')] =: \mathbf{1}[\mathbf{c} \succ \mathbf{c}'], \tag{4.7}$$

where $\Gamma : \mathcal{H} \times \mathcal{H} \to \{0, 1\}$ and $\Omega : \mathcal{C} \times \mathcal{C} \to \{0, 1\}$. The query asks whether h is preferred to h' (equivalent to $\mathbf{c}$ is preferred to $\mathbf{c}'$), as measured by ϕ.

We elicit metrics which are functions of the confusion matrix, thus comparison queries using classifiers are indistinguishable from comparison queries using confusions. Henceforth, for simplicity of notation, we denote any query as confusions based query. Next, we formally state the ME problem.

Definition 4.4 (Metric Elicitation with Pairwise Queries (given $\{(\mathbf{x}, y)_i\}_{i=1}^n$))**.** Suppose that the oracle's (unknown) performance metric is ϕ. Using oracle queries of the form $\Omega(\hat{\mathbf{c}}, \hat{\mathbf{c}}')$, where $\hat{\mathbf{c}}, \hat{\mathbf{c}}'$ are the estimated off-diagonal confusions 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 fidelity of the recovered metric and the query complexity. Given the formal definitions, we can now proceed. As is standard in the decision theory literature [6, 24], we present our ME solution by first assuming access to population quantities such as the population confusions $\mathbf{c}(h, \mathbb{P})$, then examine practical implementation by considering the estimation error from finite samples e.g. with empirical confusions $\hat{\mathbf{c}}(h, \{(\mathbf{x}, y)_i\}_{i=1}^n)$.

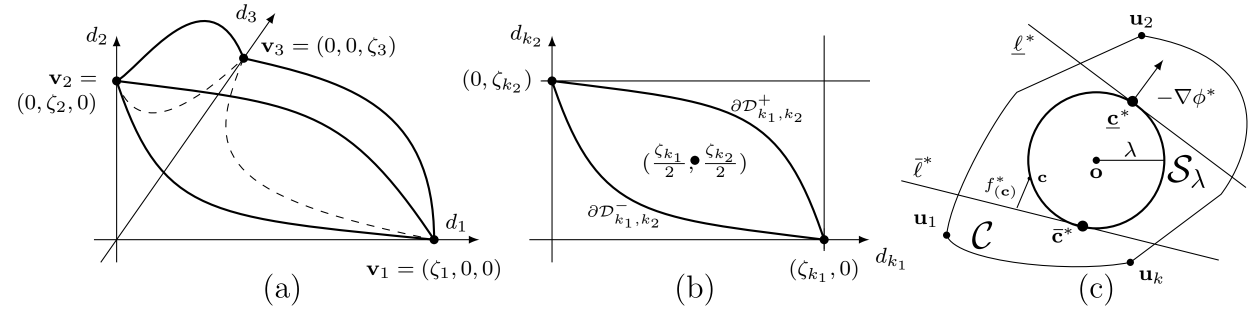


Figure 4.1: (a) Geometry of the space of diagonal confusions $\mathcal{D}$ for $k = 3$: a strictly convex space. Notice that each of the three axis-aligned faces are equivalent in geometry to the following figure in (b); (b) Geometry of diagonal confusions when restricted to classifiers predicting only classes k_1 and k_2 i.e. $\mathcal{D}_{k_1,k_2}$; (c) A sphere S_λ centered at $\mathbf{o}$ with radius λ, contained in the convex space of off-diagonal confusions $\mathcal{C}$. $f^*(\mathbf{c})$ denotes the distance of $\mathbf{c}$ from the hyperplane $\overline{\ell}^*$ tangent at $\overline{\mathbf{c}}^*$.

4.2 GEOMETRY AND PARAMETRIZATIONS OF THE QUERY SPACES

For any query based approach, it is important to understand the structure of the query space. Thus, we first study the properties of the query spaces and then develop parametrizations required for efficient elicitation. Readers may find these properties independently useful in other applications as well.

4.2.1 Geometry of the space of diagonal confusions $\mathcal{D}$ and parametrization of its boundary

Let $\mathbf{v}_i \in \mathbb{R}^k$ for $i \in [k]$ be the vectors with ζ_i at the i-th index and zero everywhere else. Notice that $\mathbf{v}_i$'s are the diagonal confusions of the trivial classifiers predicting only class i on the entire space $\mathcal{X}$.

Proposition 4.1 (Geometry of $\mathcal{D}$ – Figure 4.1 (a))**.** Under Assumption 4.1, the space of diagonal confusions $\mathcal{D}$ is strictly convex, closed, and contained in the box $[0, \zeta_1] \times \cdots \times [0, \zeta_k]$. The diagonal confusions $\mathbf{v}_i \, \forall \, i \in [k]$ are the only vertices of $\mathcal{D}$. Moreover, for any $k_1, k_2 \in [k]$, the 2-dimensional (k_1, k_2) axes-aligned face of $\mathcal{D}$ is $\mathcal{D}_{k_1,k_2}$ (Figure 4.1 (b)), which is equivalent to the space of binary classification confusion matrices confined to classes k_1, k_2. In particular, $\mathcal{D}_{k_1,k_2}$ is strictly convex.

Proposition 4.1 characterizes the geometry of the space of diagonal confusions $\mathcal{D}$. Figure 4.1(a) illustrates this geometry when $k = 3$. Interestingly, the 2-dimensional axes-aligned faces of $\mathcal{D}$ (Figure 4.1 (b)) have exactly the same geometry as the space of binary classification confusion matrices (compare this with Figure 3.1), where recall that a binary classification confusion matrix is uniquely determined by its two diagonal elements due to (4.2). We will exploit the set $\mathcal{D}_{k_1,k_2}$ (more specifically, its boundary) for the elicitation task. Now notice that for $\psi \in \varphi_{DLPM}$, the RBO classifier restricted to predict classes k_1, k_2, predicts the label (out of the two possible choices) that maximizes the expected utility conditioned on the instance. This is discussed below.

Proposition 4.2. Let $\psi \in \varphi_{DLPM}$ be parametrized by $\mathbf{a}$ such that $\|\mathbf{a}\|_1 = 1$, and let $k_1, k_2 \in [k]$, then

$$\bar{h}_{k_1,k_2}(\mathbf{x}) = \left\{ \begin{array}{ll} k_1, & \text{if } a_{k_1}\eta_{k_1}(\mathbf{x}) \geq a_{k_2}\eta_{k_2}(\mathbf{x}) \\ k_2, & \hfill o.w. \end{array} \right\} \tag{4.8}$$

is the Restricted Bayes Optimal classifier (restricted to classes k_1, k_2) with respect to ψ.

For a metric $\psi \in \varphi_{DLPM}$, Proposition 4.2 provides RBO classifiers in $\mathcal{H}_{k_1,k_2}$, which further gives us RBO diagonal confusions $\bar{\mathbf{d}}_{k_1,k_2}$ using (4.1). We know that this $\bar{\mathbf{d}}_{k_1,k_2}$ is unique, since any linear metric over a strictly convex domain ($\mathcal{D}_{k_1,k_2}$) is maximized at a unique point on the boundary [26]. So, given a DLPM, we have access to a unique point in the query space. This allows us to define and then parametrize a subset of the query space, specifically, the upper boundary of $\mathcal{D}_{k_1,k_2}$ through DLPMs.

Definition 4.5. The upper boundary of $\mathcal{D}_{k_1,k_2}$, denoted by $\partial\mathcal{D}^+_{k_1,k_2}$, constitutes the RBO diagonal confusions confined to classes $k_1, k_2 \in [k]$ for monotonically increasing DLPMs ($a_i \geq 0 \,\forall\, i \in [k]$) such that at least one out of a_{k_1} or a_{k_2} is non-zero (i.e., $a_{k_1} + a_{k_2} > 0$).

Parameterizing the upper boundary $\partial\mathcal{D}^+_{k_1,k_2}$. Let $m \in [0, 1]$. Construct a DLPM by setting $a_{k_1} = m$, $a_{k_2} = 1 - m$, and $a_i = 0$ for $i \neq k_1, k_2$. By using Proposition 4.2 and (4.1), obtain its RBO diagonal confusions, which by definition lies on the upper boundary. Thus, varying m in this process, parametrizes the upper boundary $\partial\mathcal{D}^+_{k_1,k_2}$. We denote this parametrization by $\nu(m; k_1, k_2)$, where $\nu : ([0, 1]; k_1, k_2) \to \partial\mathcal{D}^+_{k_1,k_2}$.

4.2.2 Geometry of the space $\mathcal{C}$ and parametrization of the enclosed sphere

Recall that, unlike the diagonal case, we focus on eliciting LPMs monotonically decreasing in the elements of the off-diagonal confusions (Section 4.1.2). To this end, let $\mathbf{u}_i \in \mathcal{C}$ for $i \in [k]$ be the off-diagonal confusions achieved by trivial classifiers predicting only class i on the entire space $\mathcal{X}$.

Proposition 4.3 (Geometry of $\mathcal{C}$ – Figure 4.1 (c))**.** The space of off-diagonal confusions $\mathcal{C}$ is convex and contained in the box $[0, \zeta_1]^{(k-1)} \times \cdots \times [0, \zeta_k]^{(k-1)}$. $\{\mathbf{u}_i\}_{i=1}^k$ belong to the set of vertices of $\mathcal{C}$. $\mathcal{C}$ always contains the point $\mathbf{o} = \frac{1}{k}\sum_{i=1}^{k}\mathbf{u}_i$ which corresponds to the off-diagonal confusions of the trivial classifier that randomly predicts each class with equal probability on the entire space $\mathcal{X}$.

We find that the space of off-diagonal confusions $\mathcal{C}$ has quite different geometry than the diagonal case. For instance, $\mathcal{C}$ is not strictly convex. Nevertheless, since $\mathcal{C}$ is convex and always contains the point $\mathbf{o}$, we may make the following assumption. Please see Figure 4.1(c) for an illustration.

Assumption 4.2. There exists a q-dimensional sphere $\mathcal{S}_\lambda \subset \mathcal{C}$ of radius $\lambda > 0$ centered at $\mathbf{o}$.

Such a sphere always exists as long as the class-conditional distributions are not completely overlapping, i.e., there is some signal for non-trivial classification. A method to obtain $\mathcal{S}_\lambda$ is discussed in Section 4.5. Now recall that the optimum for a linear function optimized over a sphere is given by the slope of the function scaled by the radius of the sphere. This is formalized as a trivial lemma below.

Lemma 4.1. Let $\phi \in \varphi_{LPM}$ be parametrized by $\mathbf{a}$ such that $\|\mathbf{a}\|_2 = 1$, then the unique optimal off-diagonal confusion $\bar{\mathbf{c}}$ over the sphere $\mathcal{S}_\lambda$ is a point on the boundary of $\mathcal{S}_\lambda$ given by $\bar{\mathbf{c}} = \lambda\mathbf{a} + \mathbf{o}$.

Given an LPM, Lemma 4.1 provides a unique point in the query space $\mathcal{S}_\lambda \subset \mathcal{C}$. This gives us an opportunity to characterize and then parametrize a subset of the query space through LPMs. Since we focus on eliciting monotonically decreasing LPMs, we parametrize the lower boundary of $\mathcal{S}_\lambda$.

Definition 4.6. The lower boundary of $\mathcal{S}_\lambda$, denoted by $\partial\mathcal{S}_\lambda^-$, constitutes the set of optimal off-diagonal confusions over the sphere $\mathcal{S}_\lambda$ for LPMs with $a_i \leq 0 \; \forall \, i \in [q]$ (monotonically decreasing condition).

Parameterizing the lower boundary of the enclosed sphere $\partial\mathcal{S}_\lambda^-$. We follow the standard method for parametrizing points on the surface of a sphere via angles. Let $\boldsymbol{\theta}$ be a $(q-1)$-dimensional vector of angles, where all the angles except the primary angle are in second quadrant, i.e., $\{\theta_i \in [\pi/2, \pi]\}_{i=1}^{q-2}$, and the primary angle is in the third quadrant, i.e., $\theta_{(q-1)} \in [\pi, 3\pi/2]$. Construct an LPM ($\|\mathbf{a}\|_2 = 1$) by setting $a_i = \Pi_{j=1}^{i-1}\sin\theta_j\cos\theta_i$ for $i \in [q-1]$ and $a_q = \Pi_{j=1}^{q-1}\sin\theta_j$. The choice of the quadrants ensures the monontonically decreasing condition, i.e., $\{a_i \leq 0\}_{i=1}^{q}$. By using Lemma 4.1, obtain its BO off-diagonal confusions over the sphere $\mathcal{S}_\lambda$, which clearly lies on the lower boundary. Thus, varying $\boldsymbol{\theta}$ in this procedure, parametrizes the lower boundary $\partial\mathcal{S}_\lambda^-$. We denote this parametrization by $\mu(\boldsymbol{\theta})$, where $\mu : [\pi/2, \pi]^{q-2} \times [\pi, 3\pi/2] \rightarrow \partial\mathcal{S}_\lambda^-$.

4.3 METRIC ELICITATION

Using the outlined parametrizations $\{\nu, \mu\}$, we propose efficient binary-search type algorithms to elicit oracle's implicit performance metric. We will first discuss elicitation with no *feedback* noise from the oracle. We will later show robustness to noisy feedback in Section 4.5.

4.3.1 DLPM Elicitation

The following lemma concerning a broader family of metrics is the route to our elicitation procedures. Since both linear and linear-fractional functions are quasiconcave, the lemma applies to both.

Lemma 4.2. Let $\psi : \mathcal{D} \to \mathbb{R}$ be a quasiconcave metric which is monotone increasing in all $\{d_i\}_{i=1}^k$. For $k_1, k_2 \in [k]$, let $\rho^+ : [0,1] \to \partial\mathcal{D}^+_{k_1,k_2}$ be a continuous, bijective, parametrization of the upper boundary. Then the composition $\psi \circ \rho^+ : [0,1] \to \mathbb{R}$ is quasiconcave and thus unimodal on $[0,1]$.

Remark 4.1. Under Assumption 4.1, every supporting hyperplane of $\mathcal{D}_{k_1,k_2}$ supports a unique point on the boundary $\partial\mathcal{D}^+_{k_1,k_2}$ and vice-versa (Proposition 4.1); therefore, the composition $\psi \circ \rho^+$ has no flat regions. In other words, the function $\psi \circ \rho^+$ is concave.

The proof of Lemma 4.2 first shows that any quasiconcave metric ψ defined on the space $\mathcal{D}$ is also quasiconcave on the restricted space $\mathcal{D}_{k_1,k_2}$, and then shows the quasiconcavity and thus the unimodality (due to the one-dimensional parametrization of $\partial\mathcal{D}^+_{k_1,k_2}$) of ψ on a further restricted space $\partial\mathcal{D}^+_{k_1,k_2}$. Furthermore, Remark 4.1 reveals that the function $\psi \circ \rho^+$ is concave, allowing us to devise the following binary-search type method for elicitation.

Suppose that the oracle's metric is $\psi^* \in \varphi_{DLPM}$ parametrized by $\mathbf{a}^*$ where $\|\mathbf{a}^*\|_1 = 1$, $\{a_i^*\}_{i=1}^k \geq 0$ (Section 4.1.2). Using the parametrization ν, Algorithm 4.1 returns an estimate $\hat{\mathbf{a}}$ of $\mathbf{a}^*$. It takes two classes at a time, class 1 and class i. Since the metric is unimodal on $\partial\mathcal{D}^+_{1,i}$ (Lemma 4.2), the algorithm applies binary-search in the inner while-loop to estimate the ratio a_i^*/a_1^*. The *ShrinkInterval-1* subroutine shrinks the interval $[m^a, m^b]$ into half based on the oracle responses in the usual binary-search way for searching the optimum (Figure B.1, Appendix B.1). The algorithm repeats this $(k-1)$ times to estimate the ratios $\{a_2^*/a_1^*, \ldots, a_k^*/a_1^*\}$. Finally, it outputs a normalized metric estimate $\hat{\mathbf{a}}$.

4.3.2 LPM Elicitation

We now discuss LPM elicitation, where the metrics are assumed to be monotonically decreasing in the off-diagonal confusions. Unfortunately, $\partial\mathcal{C}$ may have flat regions due to

Algorithm 4.1 DLPM Elicitation

```
Input: $\epsilon > 0$, oracle $\Omega$, $\hat{a}_1 = 1$
for $i = 2, \cdots, k$ do
    Initialize: $m^a = 0$, $m^b = 1$.
    while $\left|m^b - m^a\right| > \epsilon$ do
        Set $m^c = \frac{3m^a+m^b}{4}$, $m^d = \frac{m^a+m^b}{2}$, and $m^e = \frac{m^a+3m^b}{4}$.
        Set $\overline{\mathbf{d}}^a_{1,i} = \nu(m^a; 1, i)$ (i.e. parametrization of $\partial\mathcal{D}^+_{1,i}$ in Section 4.2.1). Similarly, set $\overline{\mathbf{d}}^c_{1,i}, \overline{\mathbf{d}}^d_{1,i}, \overline{\mathbf{d}}^e_{1,i}, \overline{\mathbf{d}}^b_{1,i}$.
        Query $\Omega(\overline{\mathbf{d}}^c_{1,i}, \overline{\mathbf{d}}^a_{1,i}), \Omega(\overline{\mathbf{d}}^d_{1,i}, \overline{\mathbf{d}}^c_{1,i})$, $\Omega(\overline{\mathbf{d}}^e_{1,i}, \overline{\mathbf{d}}^d_{1,i})$, and $\Omega(\overline{\mathbf{d}}^b_{1,i}, \overline{\mathbf{d}}^e_{1,i})$.
        $[m^a, m^b] \leftarrow$ ShrinkInterval-1 (responses).
    end while
    Set $m^d = \frac{m^a+m^b}{2}$. Then set $\hat{a}_i = \frac{1-m^d}{m^d}\hat{a}_1$.
end for
Output: $\hat{\mathbf{a}} = \left(\frac{\hat{a}_1}{\|\hat{\mathbf{a}}\|_1}, \cdots, \frac{\hat{a}_k}{\|\hat{\mathbf{a}}\|_1}\right)$.
```

lack of strict convexity, so the algorithm for the diagonal case does not apply. Instead, we consider a query space given by the sphere $\mathcal{S}_\lambda \subset \mathcal{C}$ and propose a coordinate-wise binary-search style algorithm, which is an outcome of our novel geometric characterization and the approach in Derivative-Free Optimization (DFO) [36].

Suppose that the oracle's metric is $\phi^* \in \varphi_{LPM}$ parametrized by $\mathbf{a}^*$ where $\|\mathbf{a}^*\|_2 = 1$, $\{a^*_i\}_{i=1}^q \leq 0$ (Section 4.1.2). Using the parametrization $\mu(\boldsymbol{\theta})$ of $\partial\mathcal{S}^-_\lambda$ (Section 4.2.2), Algorithm 4.2 returns an estimate $\hat{\mathbf{a}}$ of $\mathbf{a}^*$. In each iteration, the algorithm updates one angle θ_j keeping other angles fixed by a binary-search procedure, where again the *ShrinkInterval-2* subroutine shrinks the interval $[\theta^a_j, \theta^b_j]$ by half based on the oracle responses (Figure B.2, Appendix B.1). Then the algorithm cyclically updates each angle until it converges to a metric sufficiently close to the true metric. The convergence is assured because, intuitively, the algorithm via a dual interpretation minimizes a smooth, strongly convex function $f^*(\mathbf{c})$ measuring the distance of the boundary points from a hyperplane $\overline{\ell}^*$, whose slope is given by $\mathbf{a}^*$ and is tangent at the BO confusion $\bar{\mathbf{c}}^*$ (see Figure 4.1(c)).

4.4 EXTENSIONS

We emphasize that the goal of ME is not simply to choose between default or popularly used metrics but to elicit novel metrics which best match the oracle preferences. As the family of human evaluation metrics is believed to be large and since we already have created strategies for linear metrics, we can now certainly aim at efficient elicitation for flexible metric families. Therefore, in this section, we discuss a variety of extensions to other family

Algorithm 4.2 LPM Elicitation

```
Input: $\epsilon > 0$, oracle $\Omega$, $\lambda$, and $\boldsymbol{\theta} = \boldsymbol{\theta}^{(1)}$
for $t = 1, 2, \cdots, T$ do
    Set $\boldsymbol{\theta}^a = \boldsymbol{\theta}^c = \boldsymbol{\theta}^d = \boldsymbol{\theta}^e = \boldsymbol{\theta}^b = \boldsymbol{\theta}^{(t)}$.
    if $(t\%(q-1))$ then
        Set $j = t\%(q-1)$
    else
        Set $j = q-1$.
    end if
    if $j == q-1$ then
        Initialize: $\theta^a_j = \pi$, $\theta^b_j = 3\pi/2$.
    else
        Initialize: $\theta^a_j = \pi/2$, $\theta^b_j = \pi$.
    end if
    while $\left|\theta^b_j - \theta^a_j\right| > \epsilon$ do
        Set $\theta^c_j = \frac{3\theta^a_j+\theta^b_j}{4}$, $\theta^d_j = \frac{\theta^a_j+\theta^b_j}{2}$, and $\theta^e_j = \frac{\theta^a_j+3\theta^b_j}{4}$.
        Set $\bar{\mathbf{c}}^a = \mu(\boldsymbol{\theta}^a)$ (i.e. parametrization of $\partial\mathcal{S}^-_\lambda$ in Section 4.2.2) Similarly, set
            $\bar{\mathbf{c}}^c, \bar{\mathbf{c}}^d, \bar{\mathbf{c}}^e, \bar{\mathbf{c}}^b$.
        Query $\Omega(\bar{\mathbf{c}}^c, \bar{\mathbf{c}}^a), \Omega(\bar{\mathbf{c}}^d, \bar{\mathbf{c}}^c)$, $\Omega(\bar{\mathbf{c}}^e, \bar{\mathbf{c}}^d), \Omega(\bar{\mathbf{c}}^b, \bar{\mathbf{c}}^e)$
        $[\theta^a_j, \theta^b_j] \leftarrow$ ShrinkInterval-2 (responses).
    end while
    Set $\theta^d_j = \frac{1}{2}(\theta^a_j + \theta^b_j)$ and then set $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^d$.
end for
Output: $\hat{a}_i = \Pi_{j=1}^{i-1} \sin\theta_j^{(T)} \cos\theta_i{}^{(T)}$  $\forall i \in [q-1]$ and $\hat{a}_q = \Pi_{j=1}^{q-1} \sin\theta_j^{(T)}$.
```

of metrics.

For the purpose of clarity in this section, let us replace the notation of the parametrization $\nu(m; k_1, k_2)$ of the upper boundary $\partial\mathcal{D}^+_{k_1,k_2}$ by $\nu^+(m; k_1, k_2)$. This is useful to disambiguate with the parametrization $\nu^-(m; k_1, k_2)$ of the lower boundary $\partial\mathcal{D}^-_{k_1,k_2}$, which is useful in linear-fractional elicitation.

In addition to the entities defined in Table 4.1, we define some more entities such as the Inverse Bayes Optimal (IBO) and Restricted Inverse Bayes Optimal (RIBO) classifiers, diagonal confusions, utility in Table 4.2. The six definitions on the left can be analogously described diagonal metrics and diagonal confusions. The six definitions on the right are of interest for the diagonal case. These are useful in the elicitation of linear-fractional metrics.

Lastly, for linear-fractional elicitation, we need to parametrize the lower boundary $\partial\mathcal{D}^-_{k_1,k_2}$ and upper boundary of the sphere $\partial\mathcal{S}^+_\lambda$ as well. These parametrizations are defined below.

Definition 4.7. The RBO diagonal confusions for DLPMs parametrized by $\mathbf{a}$ with $a_{k_1}, a_{k_2} < 0$ form the lower boundary of $\mathcal{D}_{k_1,k_2}$, denoted by $\partial\mathcal{D}^-_{k_1,k_2}$.

Table 4.2: Bayes Optimal (BO), Inverse Bayes Optimal (IBO), Restricted Bayes Optimal (RBO), and Restricted Inverse Bayes Optimal (RIBO) entities.

Name	Definition	Name	Definition
BO classifier $\overline{h}$	$\text{argmax}_{h \in \mathcal{H}}\, \phi(\mathbf{c}(h))$	RBO classifier $\overline{h}_{k_1,k_2}$	$\text{argmax}_{h \in \mathcal{H}_{k_1,k_2}}\, \psi(\mathbf{d}(h))$
BO utility $\overline{\tau}$ over a subset $\mathcal{S} \subseteq \mathcal{C}$	$\max_{\mathbf{c} \in \mathcal{S} \subseteq \mathcal{C}} \phi(\mathbf{c})$	RBO utility $\overline{\tau}_{k_1,k_2}$	$\max_{\mathbf{d} \in \mathcal{D}_{k_1,k_2}} \psi(\mathbf{d})$
BO confusion $\bar{\mathbf{c}}$ over a subset $\mathcal{S} \subseteq \mathcal{C}$	$\underset{\mathbf{c} \in \mathcal{S} \subseteq \mathcal{C}}{\text{argmax}}\, \phi(\mathbf{c})$	RBO confusion $\overline{\mathbf{d}}_{k_1,k_2}$	$\underset{\mathbf{d} \in \mathcal{D}_{k_1,k_2}}{\text{argmax}}\, \psi(\mathbf{d})$
IBO classifier $\underline{h}$	$\text{argmin}_{h \in \mathcal{H}}\, \phi(\mathbf{c}(h))$	RIBO classifier $\underline{h}_{k_1,k_2}$	$\text{argmin}_{h \in \mathcal{H}_{k_1,k_2}}\, \psi(\mathbf{d}(h))$
IBO utility $\underline{\tau}$ over a subset $\mathcal{S} \subseteq \mathcal{C}$	$\min_{\mathbf{c} \in \mathcal{S} \subseteq \mathcal{C}} \phi(\mathbf{c})$	RIBO utility $\underline{\tau}_{k_1,k_2}$	$\min_{\mathbf{d} \in \mathcal{D}_{k_1,k_2}} \psi(\mathbf{d})$
IBO confusion $\underline{\mathbf{c}}$ over a subset $\mathcal{S} \subseteq \mathcal{C}$	$\underset{\mathbf{c} \in \mathcal{S} \subseteq \mathcal{C}}{\text{argmin}}\, \phi(\mathbf{c})$	RIBO confusion $\underline{\mathbf{d}}_{k_1,k_2}$	$\underset{\mathbf{d} \in \mathcal{D}_{k_1,k_2}}{\text{argmin}}\, \psi(\mathbf{d})$

Parametrization of $\partial\mathcal{D}^-_{k_1,k_2}$. We denote this parametrization by a function $\nu^-(m; k_1, k_2)$. Take a parameter $-1 \leq m \leq 0$. Create a DLPM ψ by setting $a_{k_1} = m$, $a_{k_2} = -1 - m$, and $a_i = 0$ for $i \neq k_1, k_2 \in [k]$. RBO diagonal confusions of such DLPMs lie on the lower boundary $\partial\mathcal{D}^-_{k_1,k_2}$. As we vary m, we move on the lower boundary $\partial\mathcal{D}^-_{k_1,k_2}$.

Definition 4.8. The optimal off-diagonal confusions over the sphere S_λ for LPMs parametrized by $\mathbf{a}$ with $a_i \geq 0 \, \forall \, i \in [k]$ form the upper boundary of S_λ, denoted by ∂S^+_λ.

Parametrization of ∂S^+_λ. The parametrization of the upper boundary ∂S^+_λ is same as that of the lower boundary ∂S^-_λ (Section 4.2.2) except that now all the angles are in the first quadrant i.e. $\{\theta_i \in [0, \pi/2]\}_{i=1}^{q-1}$, so to satisfy the condition $a_i \geq 0 \, \forall \, i \in [k]$.

4.4.1 Diagonal Linear Fractional Performance Metric (DLFPM) Elicitation

We start by first defining the diagonal linear fractional performance metric.

Definition 4.9. Diagonal Linear-Fractional Performance Metric (DLFPM): We denote this family by φ_{DLFPM}. Given $\mathbf{a}, \mathbf{b} \in \mathbb{R}^k$ and $b_0 \in \mathbb{R}$, the metric is defined as:

$$\psi(\mathbf{d}) = \frac{\langle \mathbf{a}, \mathbf{d} \rangle}{\langle \mathbf{b}, \mathbf{d} \rangle + b_0}. \tag{4.9}$$

For any $\psi \in \varphi_{DLFPM}$, we assume that $\{a_i\}_{i=1}^k, \{b_i\}_{i=1}^k$ are not all zero simultaneously and wlog, we take $\psi(\mathbf{d}) \in [0, 1]$ and monotonically increasing in all $\{d_i\}_{i=1}^k$. We also make the following regularity assumption.

Assumption 4.3. Let $\psi \in \varphi_{DLFPM}$ parametrized by $\mathbf{a}$ and $\mathbf{b}$ (Definition 4.9). We assume that $a_i \geq 0$ and $a_i \geq b_i$ for all $i \in [k]$. In addition, $b_0 = \sum_i (a_i - b_i)\zeta_i$ and $\sum_i a_i = 1$.

Equivalent to fixing $\|\mathbf{a}\|_1 = 1$, $a_i \geq 0$ for the diagonal linear case (Section 4.1.2), the conditions in Assumption 4.3 are sufficient conditions for DLFPMs to be bounded and monotonically increasing in diagonal elements of the confusion matrices. This is detailed in the following proposition.

Proposition 4.4. The conditions in Assumption 4.3 are sufficient for $\psi \in \varphi_{DLFPM}$ to be bounded in $[0, 1]$ and simultaneously monotonically increasing in $\{d_i\}_{i=1}^k$.

We consider $b_0 = \sum_i (a_i - b_i)\zeta_i$, instead of the derived condition $b_0 \geq \sum_i (a_i - b_i)\zeta_i$, which is sufficient to guarantee a unique metric bounded in $[0, 1]$ for elicitation purposes (instead of one of the equivalent alternatives). Note that most existing linear-fractional metrics satisfy these conditions [6, 15, 24].

Now, suppose that the oracle's metric is $\psi^* \in \varphi_{DLFPM}$. Let $\overline{\tau}^*$ and $\underline{\tau}^*$ be the maximum and minimum value of ψ^*, respectively. Due to strict convexity of $\mathcal{D}$, we have a hyperplane

$$\overline{\ell}_f^* := \sum_{i=1}^k (a_i^* - \overline{\tau}^* b_i^*) d_i^* = \overline{\tau}^* b_0 \tag{4.10}$$

tangent at the BO diagonal confusions $\overline{\mathbf{d}}^*$ on the upper boundary of $\mathcal{D}$, denoted by $\partial\mathcal{D}^+$. Similarly, we have a hyperplane

$$\underline{\ell}_f^* := \sum_{i=1}^k (a_i^* - \underline{\tau}^* b_i^*) \underline{d}_i^* = \underline{\tau}^* b_0 \tag{4.11}$$

which touches the set $\mathcal{D}$ only at $\underline{\mathbf{d}}^*$ (IBO diagonal confusions) on the lower boundary, denoted by $\partial\mathcal{D}^-$. See Figure 4.1(c) for the visual intuition, where assume that the underlying space is $\mathcal{D}$ instead of the sphere $\mathcal{S}_\lambda$.

Since DLFPM is quasiconcave, Algorithm 4.1 returns a slope of the hyperplane, say $\bar{\mathbf{s}}$. Using that slope, we can compute the Bayes Optimal diagonal confusions $\overline{\mathbf{d}}^*$ using Proposition B.1 (a more general version of Proposition 4.2), which gives us the hyperplane $\overline{\ell}^* := \langle \bar{\mathbf{s}}, \mathbf{d} \rangle = \langle \bar{\mathbf{s}}, \overline{\mathbf{d}}^* \rangle$. This is equivalent to $\overline{\ell}_f^*$ up to a constant multiple; therefore, the true metric is the solution to the following non-linear system of equations (SoE):

$$a_i^* - \overline{\tau}^* b_i^* = \alpha \bar{s}_i \;\; \forall\, i \in [k], \quad \overline{\tau}^* b_0^* = \alpha \langle \bar{\mathbf{s}}, \overline{\mathbf{d}}^* \rangle \tag{4.12}$$

where $\alpha \geq 0$, because LHS and $\bar{s}_i$'s are non-negative. If we somehow know the true $\mathbf{a}^*$, then

Algorithm 4.3 Diagonal (Quasiconcave) Metric Minimization

```
Follow Algorithm 4.1 except:
**Initialize:** $m^a = -1$, $m^b = 0$ in step 3 of Algorithm 4.1.
**Invert Responses:** Replace oracle responses $\mathbf{d} \prec \mathbf{d}'$ with $\mathbf{d} \succ \mathbf{d}'$ and vice versa.
```

by using the following proposition, we can elicit the DLFPM upto a constant multiple, i.e. we can get $\hat{\psi} \approx \alpha\psi^*$, which is sufficient for the elicitation task.

Proposition 4.5. Knowing $\mathbf{a}^*$ i.e. using $\hat{\mathbf{a}} = \mathbf{a}^*$ solves the SoEs (4.12) as:

$$\hat{b}_i = (\hat{a}_i - \bar{s}_i)\frac{\Lambda_1}{\Lambda_2}, \tag{4.13}$$

where $\Lambda_1 = \sum_i \hat{a}_i\zeta_i$, $\Lambda_2 = \langle \bar{\mathbf{s}}, \overline{\mathbf{d}}^* \rangle + \sum_i(\hat{a}_i - \bar{s}_i)\zeta_i$, and $\hat{b}_0$ is as defined in Assumption 4.3.

Now the question is how do we get the true $\mathbf{a}^*$. To our rescue, we also know that a DLFPM is quasiconvex. Thus, by minimizing the metric (again by using restricted classifiers) using Algorithm 4.3 (described next), we can get a similar hyperplane on the lower boundary $\partial\mathcal{D}^-$. Algorithm 4.3 is described below.

Algorithm 4.3. *Minimizing diagonal quasiconvex metrics:* This algorithm is same as Algorithm 4.1 with only two changes. First, we start with $m \in [-1, 0]$, because the optimum will lie on the lower boundary $\partial\mathcal{D}^-$. Second, we check for $\mathbf{d} \prec \mathbf{d}'$ whenever Algorithm 4.1 checks for $\mathbf{d} \succ \mathbf{d}'$, and vice-versa. Here, we output the counterpart, i.e., slope $\underline{\mathbf{s}}$.

Once we get the slope $\underline{\mathbf{s}}$, we can obtain the inverse Bayes diagonal confusion $\underline{\mathbf{d}}^*$ using Proposition B.1 (a more general version of Proposition 4.2). This will result in a supporting hyperplane $\underline{\ell}^* := \langle \underline{\mathbf{s}}, \mathbf{d} \rangle = \langle \underline{\mathbf{s}}, \underline{\mathbf{d}}^* \rangle$. This hyperplane is tangent to the lower boundary $\partial\mathcal{D}^-$, and equivalent to $\underline{\ell}_f^*$ up to a constant multiple; thus, the true metric is also the solution of the following SoE:

$$a_i^* - \underline{\tau}^* b_i^* = \gamma \underline{s}_i \ \ \forall\, i \in [k], \quad \underline{\tau}^* b_0^* = \gamma \langle \underline{\mathbf{s}}, \underline{\mathbf{d}}^* \rangle \tag{4.14}$$

where $\gamma \leq 0$ since LHS is positive, but $\underline{s}_i$'s are negative. Again, we may assume $\gamma < 0$. By dividing the above equations by $-\gamma$ on both sides, all the coefficients are factored by $-\gamma$. This does not change ψ^*; thus, the system of equations becomes the following:

$$a_i'' - \underline{\tau}^* b_i'' = \underline{s}_i, \ \ \forall\, i \in [k], \quad \underline{\tau}^* b_0'' = \langle \underline{\mathbf{s}}, \underline{\mathbf{d}}^* \rangle. \tag{4.15}$$

Now, if we know $\mathbf{a}'$ in (B.19), then by using Proposition 4.5, we may solve the system (B.19) and obtain a metric, say ψ'. System (4.15) can be solved analogously, provided we know

Algorithm 4.4 DLFPM: Grid Search for Best Pairwise Ratios

```
Input: $n', \delta$.
for $j = 2, \cdots, k$ do
    Initialize: $\sigma_{opt} = \infty, a'_j = 0$.
    Sample $\mathbf{d}^1, ..., \mathbf{d}^{n'}$ on $\partial\mathcal{D}_{1,j}$ (BO or IBO diagonal confusions for random $n'$ DLPMs).
    for ($a'_j = 0$; $a'_j \leq 1$; $a'_j = a'_j + \delta$) do
        Compute $\psi'$, $\psi''$ using Proposition 4.5.
        Compute array $r = [\frac{\psi'(\mathbf{d}^1)}{\psi''(\mathbf{d}^1)}, ..., \frac{\psi'(\mathbf{d}^{n'})}{\psi''(\mathbf{d}^{n'})}]$. Set $\sigma = \text{std}(r)$.
        if ($\sigma < \sigma_{opt}$) Set $\sigma_{opt} = \sigma$ and $a'_{j,opt} = a'_j$.
    end for
    Set $a'_j = \frac{a'_{j,opt}}{1 - a'_{j,opt}}$.
end for
$a'_1 = 1$.
Output: $\mathbf{a}' = \left(\frac{a'_1}{\|\mathbf{a}'\|_1}, \cdots, \frac{a'_k}{\|\mathbf{a}'\|_1}\right)$.
```

$\mathbf{a}''$ in (4.15), to get a metric, say ψ''. Notice that when when we have the true ratio i.e $a^*_i/a^*_j = a'_i/a'_j = a''_i/a''_j$ for $i, j \in [k]$, then $\psi^* = \psi'/\alpha = -\psi''/\gamma$. This means that when the true ratios are known, then ψ', ψ'' are constant multiples of each other. So, we look for the ratios where the solution to the two systems are just pointwise constant multiple of one another. This is the same idea used in the binary case (see Section 3.4.2). However, we have to search for the entire grid $[0, 1]^k$ instead of $[0, 1]$ as is in the binary case. This is a computationally challenging task.

Notice that we can randomly sample diagonal confusions on the boundary $\partial\mathcal{D}$. This is done by first randomly generating DLPMs and then computing their BO or IBO diagonal confusions using Proposition B.1. After obtaining $\overline{\ell}^*$ and $\underline{\ell}^*$, we run the grid seacrh based Algorithm 4.4 to find the estimates of the true a_i's. Although the grid-search based algorithm is independent of oracle queries, it is computationally efficient. It runs for $(k - 1)$ rounds, where in each round it matches the solution of the two SoE's as closely as possible on a number of samples from the boundary $\partial\mathcal{D}_{1,k}$ and figures out the ratio of a_j/a_1 for $j \neq 1 \in [k]$. Thanks to the property $\sum_i a_i = 1$ and access to the restricted diagonal confusions, we are saved from searching the entire grid $[0, 1]^k$ to merely $(k - 1)$ times grid-search on $[0, 1]$.

4.4.2 LFPM Elicitation

We start by defining the linear-fractional performance metric in off-diagonal confusions.

Definition 4.10. Linear-Fractional Performance Metric (LFPM): We denote this family by

φ_{LFPM}. Given constants $\mathbf{a}, \mathbf{b} \in \mathbb{R}^q$ and $b_0 \in \mathbb{R}$, the metric is defined as

$$\phi(\mathbf{c}) = \frac{\langle \mathbf{a}, \mathbf{c} \rangle}{\langle \mathbf{b}, \mathbf{c} \rangle + b_0}. \tag{4.16}$$

For any $\phi \in \varphi_{LFPM}$ (Definition 4.10), we assume that $\{a_i\}_{i=1}^q, \{b_i\}_{i=1}^q$ are not all zero simultaneously. Moroever, w.l.o.g., $\phi(\mathbf{c}) \in [-1, 0]\ \forall\ \mathbf{c} \in \mathcal{C}$ and is monotonically decreasing in all $\{c_i\}_{i=1}^q$. Similar to the diagonal case, we make the following regularity assumption.

Assumption 4.4. Let $\phi \in \varphi_{LFPM}$ (Definition 4.10). We assume that $a_i \leq 0$ and $a_i \leq -b_i$ for all $i \in [q]$. In addition, $b_0 = \sum_i -(a_i + b_i)\zeta_i$, and $\sum_i a_i = -1$.

Equivalent to fixing $\|\mathbf{a}\|_1 = 1$, $a_i \geq 0$ for the diagonal linear case (Section 4.1.2), the conditions in Assumption 4.4 are sufficient conditions for LFPMs to be bounded and monotonically decreasing in off-diagonal elements of the confusion matrices. This is detailed in the following proposition.

Proposition 4.6. Assumption 4.4 is sufficient for $\phi \in \varphi_{LFPM}$ to be bounded in $[-1, 0]$ and simultaneously monotonically decreasing in $\{c_i\}_{i=1}^q$.

We consider $b_0 = \sum_i -(a_i + b_i)\zeta_i$, instead of the derived condition $b_0 \geq \sum_i -(a_i + b_i)\zeta_i$, which is sufficient to guarantee a unique metric bounded in $[-1, 0]$ for elicitation purposes (instead of one of the equivalent alternatives). Note that most existing linear-fractional metrics satisfy these conditions [6, 15, 24].

Now, suppose that the oracle's metric is $\phi^* \in \varphi_{LFPM}$. Let $\overline{\tau}^*$ and $\underline{\tau}^*$ be the maximum and minimum value of ϕ^*, respectively. Due to strict convexity of $\mathcal{S}_\lambda$, we have a hyperplane

$$\overline{\ell}_f^* := \sum_{i=1}^q (a_i^* - \overline{\tau}^* b_i^*)\overline{c}_i^* = \overline{\tau}^* b_0 \tag{4.17}$$

touching the set $\mathcal{S}_\lambda$ only at BO confusions $\bar{\mathbf{c}}^*$ (over the sphere $\mathcal{S}_\lambda$) on the lower boundary $\partial\mathcal{S}_\lambda^-$. Similarly, we have a hyperplane

$$\underline{\ell}_f^* := \sum_{i=1}^q (a_i^* - \underline{\tau}^* b_i^*)\underline{c}_i^* = \underline{\tau}^* b_0 \tag{4.18}$$

which touches the set $\mathcal{S}_\lambda$ only at inverse Bayes Optimal confusions $\underline{\mathbf{c}}^*$ (over the sphere $\mathcal{S}_\lambda$) on the upper boundary $\partial\mathcal{S}_\lambda^+$. See Figure 4.1(c) for the visual intuition.

Here, we use strict convexity of $\mathcal{S}_\lambda$ and follow the same arguments as in DLFPM to get a hyerplane $\overline{\ell}^* := \langle \bar{\mathbf{s}}, \mathbf{c} \rangle = \langle \bar{\mathbf{s}}, \bar{\mathbf{c}}^* \rangle$ after using Algortihm 4.2. Here, $\bar{\mathbf{c}}^*$ is the optimal best (BO)

Algorithm 4.5 General (Quasiconcave) Metric Minimization

```
Follow Algorithm 4.2 except:
**Initialize:** $\theta_j^a = 0$, $\theta_j^b = \pi/2$ in steps 9-13 of Algorithm 4.2.
**Invert Responses:** Replace oracle responses $\mathbf{c} \prec \mathbf{c}'$ with $\mathbf{c} \succ \mathbf{c}'$ and vice versa.
```

off-diagonal confusion on the sphere. The only difference is that the BO confusions lie on the lower boundary $\partial\mathcal{S}_\lambda^-$ (monotonically decreasing). The SoE we get is:

$$a_i^* - \bar{\tau}^* b_i^* = \alpha \bar{s}_i \;\; \forall\, i \in [q], \qquad \bar{\tau}^* b_0^* = \alpha \langle \bar{\mathbf{s}}, \bar{\mathbf{c}}^* \rangle \tag{4.19}$$

where $\alpha \geq 0$. Similar to DLFPMs, by knowing $\mathbf{a}^*$, we can elicit the LFPM upto a constant multiple.

Proposition 4.7. Knowing $\mathbf{a}^*$ i.e. using $\hat{\mathbf{a}} = \mathbf{a}^*$ solves the SoEs (4.19) as:

$$\hat{b}_i = (\hat{a}_i - \bar{s}_i)\frac{\Lambda_1'}{\Lambda_2'}, \tag{4.20}$$

where $\Lambda_1' = -\sum_i \hat{a}_i \zeta_i$, $\Lambda_2' = \langle \bar{\mathbf{s}}, \bar{\mathbf{c}}^* \rangle + \sum_i (\hat{a}_i - \bar{s}_i)\zeta_i$, and $\hat{b}_0$ is as defined in Assumption 4.4.

Now again the question is how do we get the true $\mathbf{a}^*$. To our rescue, we also know that an LFPM is quasiconvex. Thus, by minimizing the metric using Algorithm 4.5 (described next), we can get a similar hyperplane $\underline{\ell}^* := \langle \underline{\mathbf{s}}, \underline{\mathbf{c}} \rangle = \langle \underline{\mathbf{s}}, \underline{\mathbf{c}}^* \rangle$ tangent to the upper boundary $\partial\mathcal{S}_\lambda^+$.

Algorithm 4.5 *Minimizing quasiconvex metrics of off-diagonal confusions:* This algorithm is same as Algorithm 4.2 with only two changes. First, we start with $\boldsymbol{\theta} \in [0, \pi/2]^q$, because the optimum will lie on the upper boundary $\partial\mathcal{S}_\lambda^+$. Second, we check for $\mathbf{c} \prec \mathbf{c}'$ whenever Algorithm 4.2 checks for $\mathbf{c} \succ \mathbf{c}'$, and vice versa. Here, we output the counterpart, i.e., slope $\underline{\mathbf{s}}$.

Thus, a similar SoE (4.19) whose solution looks like Proposition 4.7 is obtained. After obtaining $\bar{\ell}^*$ and $\underline{\ell}^*$, we run grid-search Algorithm 4.6 to find the estimates of the true a_i's. The algebra related to LFPM elicitation is same as the DLFPM case. However, this time we need to search in $[0, 1]^{q-1}$ grid. Again, we have easy access to off-diagonal confusions on the sphere $\partial\mathcal{S}_\lambda$ corresponding to BO or IBO off-diagonal confusions for different LPMs (Lemma 4.1); therefore, we can use the following algorithm, which is analogous to Algorithm 4.4.

Algorithm 4.6 *LFPM: grid-search for best pairwise ratios:* This is same as Algorithm 4.4 except the following two changes. First, the second line of Algorithm 4.4 will have a for loop

Algorithm 4.6 LFPM: Grid-Search for Best Pairwise Ratios

```
1: Follow Algorithm 4.4 except:
2: Run the for loop in step 2 of Algorithm 4.4 for 2 to $q-1$.
3: Generate samples from $\partial\mathcal{S}_\lambda$.
```

running from 2 to $q-1$. Second, in line 4, samples will be generated from the surface of the sphere $\partial\mathcal{S}_\lambda$ as discussed above, instead of $\partial\mathcal{D}_{1,k}$.

4.4.3 Monotonic Metrics of diagonal confusions

Recall that the space $\mathcal{D}$ is strictly convex. Suppose that the oracle's metric is ψ^*, which is just monotonic increasing in $\{d_i\}_{i=1}^k$. Let $\mathbf{a}^*$ be the slope of the supporting hyperplane at the optimal diagonal confusions $\mathbf{d}^*$. Then we may use Algorithm 4.1 which will return a linear metric $\hat{\mathbf{a}}$ by using pairwise comparisons. Notice that, we may then compute an estimate of the BO diagonal confusions $\hat{\mathbf{d}}$ using Proposition B.1 corresponding to the output $\hat{\mathbf{a}}$ of the algorithm. Since the space $\mathcal{D}$ is strictly convex, $\langle \hat{\mathbf{a}}, \mathbf{d} \rangle = \langle \hat{\mathbf{a}}, \hat{\mathbf{d}} \rangle$ becomes the estimate of the unique supporting hyperplane at $\hat{\mathbf{d}}$.

The first order approximation of ψ^* at $\hat{\mathbf{d}}$ can be given by:

$$\psi^*(\mathbf{d}) = \psi^*(\hat{\mathbf{d}}) + \langle \hat{\mathbf{a}}, \mathbf{d} - \hat{\mathbf{d}} \rangle. \tag{4.21}$$

Since performance metrics are not affected by scale and additive biases, then the first order approximation given by $\langle \hat{\mathbf{a}}, \mathbf{d} \rangle$ suffices for the elicitation task. Notice that this is of high practical importance to practitioners, since this is an estimate of the weighted accuracy at the estimate of the optimal diagonal confusions.

4.5 GUARANTEES

We discuss robustness under the following feedback model, which is useful in practical scenarios, and is borrowed from Definition 2.4.

Definition 4.11 (Oracle Feedback Noise: $\epsilon_\Omega \geq 0$)**.** The oracle responds correctly as long as $|\phi(\mathbf{c}) - \phi(\mathbf{c}')| > \epsilon_\Omega$ (analogously $|\psi(\mathbf{d}) - \psi(\mathbf{d}')| > \epsilon_\Omega$). Otherwise, it may provide incorrect answers.

In other words, the oracle may respond incorrectly if the confusions are too close as measured by the metric ϕ (analogously ψ). Next, we discuss elicitation guarantees for DLPM and LPM elicitation.

Theorem 4.1. Given $\epsilon, \epsilon_\Omega \geq 0$, and a 1-Lipschitz DLPM ψ^* parametrized by $\mathbf{a}^*$. Then the output $\hat{\mathbf{a}}$ of Algorithm 4.1 after $O((k-1)\log\frac{1}{\epsilon})$ queries to the oracle satisfies $\|\mathbf{a}^* - \hat{\mathbf{a}}\|_\infty \leq O(\epsilon + \sqrt{\epsilon_\Omega})$, which is equivalent to $\|\mathbf{a}^* - \hat{\mathbf{a}}\|_2 \leq O(\sqrt{k}(\epsilon + \sqrt{\epsilon_\Omega}))$ using standard norm bounds.

Next, we guarantee LPM elicitation when the sphere radius dominates the oracle noise.

Theorem 4.2. Given $\epsilon, \epsilon_\Omega \geq 0$, and a 1-Lipschitz LPM ϕ^* parametrized by $\mathbf{a}^*$. Suppose $\lambda \gg \epsilon_\Omega$, then the output $\hat{\mathbf{a}}$ of Algorithm 4.2 after $O\left(z_1 \log(z_2/(q\epsilon^2))(q-1)\log\frac{\pi}{2\epsilon}\right)$ queries satisfies $\|\mathbf{a}^* - \hat{\mathbf{a}}\|_2 \leq O(\sqrt{q}(\epsilon + \sqrt{\epsilon_\Omega/\lambda}))$, where z_1, z_2 are constants independent of ϵ and q.

We see that the algorithms are robust to noise, and their query complexity depends linearly in the unknown entities. The term $z_1 \log(z_2/(q\epsilon^2))$ may attribute to the number of cycles in Algorithm 4.2, but due to the curvature of the sphere, we observe that it is not a dominating factor in the query complexity. For instance, we find that when $\epsilon = 10^{-2}$, two cycles (i.e. $T = 2(q-1)$ in Algorithm 4.2) are sufficient for achieving elicitation up to the error tolerance $\sqrt{q}\epsilon$. Moreover, the query complexity in Theorem 4.2 is optimal. We show this in Chapter 6 for the quadratic elicitation case, which in turn applies to the above linear elicitation case as well. One remaining question for LPM elicitation is to select a sufficiently large value of λ. Algorithm B.1 (Appendix B.4.1) provides an offline procedure to compute a $\lambda \geq \tilde{r}/k$, where $\tilde{r}$ is the radius of the largest ball contained in the set $\mathcal{C}$.

ME with Finite Samples: As a final step, we consider the following questions when working with finite samples: (a) do we get the correct feedback from querying $\Omega(\hat{\mathbf{c}}, \hat{\mathbf{c}}')$ instead of querying $\Omega(\mathbf{c}, \mathbf{c}')$? (b) what is the effect of $\hat{\eta}_i$'s when used in place of true η_i's? The answers are straightforward. Since the sample estimates of confusion matrices are consistent estimators and the metrics discussed are 1-Lipschitz with respect to the confusion matrices, with high probability, we gather correct oracle feedback as long as we have sufficient samples. Furthermore, subject to regularity assumptions, Lemma 3.3 shows that the errors due to using $\hat{\eta}$ affect the (binary) confusion matrices on the boundary in a controlled manner. Since Algorithm 4.1 uses pairwise RBO (binary) classifiers, it inherits the error guarantees in the multiclass case. On the other hand, since Algorithm 4.2 does not use the boundary, its results are agnostic to finite sample error as long as the sphere is contained within $\mathcal{C}$.

4.6 EXPERIMENTS

In this section, we empirically validate the results of theorems 4.1 and 4.2 and investigate sensitivity due to finite sample estimates.[1] For the ease of judgments, we show results for $k = 3$ and $k = 4$ classes.

[1] A subset of results is shown here. Refer Appendix B.6 for more results.

Table 4.3: DLPM elicitation at $\epsilon = 0.01$ for synthetic data. The number of queries used for $k = 3$ and $k = 4$ is 56 and 84, respectively.

Classes $k = 3$		Classes $k = 4$	
$\psi^* = \mathbf{a}^*$	$\hat{\psi} = \hat{\mathbf{a}}$	$\psi^* = \mathbf{a}^*$	$\hat{\psi} = \hat{\mathbf{a}}$
(0.21, 0.59, 0.20)	(0.21, 0.60, 0.20)	(0.22, 0.13, 0.14, 0.52)	(0.22, 0.13, 0.14, 0.52)
(0.23, 0.15, 0.62)	(0.23, 0.15, 0.62)	(0.58, 0.17, 0.08, 0.18)	(0.58, 0.17, 0.08, 0.18)

Table 4.4: LPM elicitation at $\epsilon = 0.01$ for synthetic data. The number of queries used for $k = 3$ and $k = 4$ is 320 and 704, respectively.

Classes	$\phi^* = \mathbf{a}^*$	$\hat{\phi} = \hat{\mathbf{a}}$
3	(-0.37, -0.89, -0.09, -0.23, -0.04, -0.03)	(-0.37, -0.89, -0.09, -0.23, -0.04, -0.03)
3	(-0.80, -0.55, -0.18, -0.08, -0.14, -0.05)	(-0.80, -0.55, -0.18, -0.08, -0.14, -0.05)
4	(-0.90, -0.28 -0.10, -0.31, -0.04, -0.05, -0.03, -0.04, -0.02, -0.01, -0.01, -0.01)	(-0.90, -0.28, -0.10, -0.31, -0.04, -0.05, -0.03, -0.04, -0.02, -0.01, -0.01, -0.01)
4	(-0.54, -0.10, -0.62, -0.52, -0.03, -0.07, -0.11, -0.07, -0.14, -0.03, -0.03, -0.04)	(-0.55, -0.11, -0.62, -0.51, -0.03, -0.07, -0.11, -0.07, -0.14, -0.03, -0.03, -0.04)

4.6.1 Synthetic Data Experiments

We assume a joint distribution for $\mathcal{X} = [-1, 1]$ and $\mathcal{Y} = [k]$. This is given by the marginal distribution $f_X = \mathbb{U}[-1, 1]$ and $\eta_i(x) = \frac{1}{1+e^{p_i x}}$ for $i \in [k]$, where $\mathbb{U}[-1, 1]$ is the uniform distribution on $[-1, 1]$ and $\{p_i\}_{i=1}^k$ are the parameters controlling the degree of noise in the labels. We fix $(p_1, p_2, p_3) = (1, 3, 5)$ and $(p_1, p_2, p_3, p_4) = (1, 3, 6, 10)$ for experiments with three and four classes, respectively. To verify elicitation, we first define a true metric ψ^* or ϕ^*. This specifies the query outputs of Algorithm 4.1 or Algorithm 4.2. Then we run the algorithms to check whether or not we recover the same metric. Some results are shown in Table 4.3 and Table 4.4. Results verify that we elicit the true metrics even for small $\epsilon = 0.01$, and as predicted, this requires only $4(k-1)\lceil \log(1/\epsilon) \rceil$ and $4T\lceil \log(\pi/2\epsilon) \rceil$ queries for DLPM and LPM elicitation respectively, where $\lceil \cdot \rceil$ is the ceil function and $T = 2(q-1)$.

4.6.2 Real-World Data Experiments

Finite samples may affect the size of the sphere S_λ in LPM elicitation, but we observe that as long as λ is greater than ϵ_Ω LPMs can be elicited (Appendix B.6.2). Thus, here we emprically validate only DLPM elicitation with finite samples. We consider two real-world datasets: (a) SensIT (Acoustic) dataset [37] (78823 instances, 3 classes), and (b) Vehicle dataset [38] (846 instances, 4 classes). From each dataset, we create two other datasets containing randomly chosen 50% and 75% of the datapoints. So, we have six datasets in

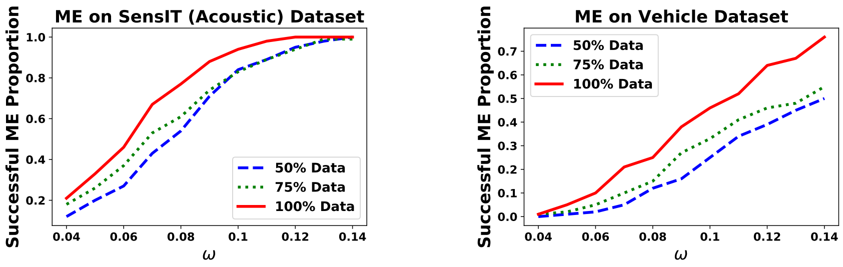


Figure 4.2: DLPM elicitation on real data for $\epsilon = 0.01$. For randomly chosen hundred $\mathbf{a}^*$, we show the proportion of times our estimates $\hat{\mathbf{a}}$ obtained with $4(k-1)\lceil\log(1/\epsilon)\rceil$ queries satisfy $\|\mathbf{a}^* - \hat{\mathbf{a}}\|_\infty \leq \omega$.

total. For all the datasets, we standardize the features and split the dataset into two parts $\mathcal{S}_1$ and $\mathcal{S}_2$. On $\mathcal{S}_1$, we learn $\{\hat{\eta}_i(x)\}_{i=1}^k$ using a regularized softmax regression model. We use $\mathcal{S}_2$ for making predictions and computing sample confusions.

We randomly selected 100 DLPMs i.e. $\mathbf{a}^*$'s. We then used Algorithm 4.1 with $\epsilon = 0.01$ to recover the estimates $\hat{\mathbf{a}}$'s. In Figure 4.2, we show the proportion of times $\|\mathbf{a}^* - \hat{\mathbf{a}}\|_\infty \leq \omega$ for different values of ω. We see improved elicitation as we increase the number of datapoints in both the datasets, suggesting that ME improves with larger datasets. In particular, for the full SensIT (Acoustic) dataset, we elicit all the metrics within $\omega = 0.12$. We also observe that $\omega \in [0.04, 0.08]$ is an overly tight evaluation criterion that can result in failures. This is because the elicitation routine gets stuck at the closest achievable sample confusions, which need not be optimal within the (small) search tolerance ϵ.

4.7 DISCUSSION AND FUTURE WORK

- **Practical Convenience.** Our procedures can also be applied by posing pairwise classifier comparisons directly. One way is to use A/B testing [21] where the user population acts an oracle. Another way is to use comparisons from a single expert, perhaps combined with interpretable machine learning techniques [19, 20]. We suggest the approach proposed by Narasimhan [15] for estimating the classifier associated with a given confusion matrix.
- **Advantage of Algorithm 4.1.** If there is a reason to restrict the metric search to DLPM e.g. due to prior knowledge, then Algorithm 4.1 is preferred for its lower query complexity.
- **Future Work.** We plan to extend our procedures for the oracles that are only probably correct. This can be done easily by applying majority voting over repeated queries [39].

4.8 RELATED WORK

The closest line of work to this chapter is the simpler setting of binary classification from Chapter 3. As we move to multiclass performance ME, we find that the form of metrics and the complexity of the query space increases. This results in stark differences in the elicitation algorithms. Algorithm 4.1, which is closest to the binary approach, only works for Restricted Bayes Optimal classifiers, and Algorithm 4.2 requires a coordinate-wise binary-search approach. As a result, novel methods are also required to provide query complexity guarantees. The LPM elicitation problem can be posed as a Derivative-Free Optimization [36] to a certain extent, but only after exploiting the geometry as we have. In addition, passively learning linear functions using pairwise comparisons has been studied before [31, 34, 40], but these approaches fail to control sample (i.e. query) complexity and end up utilizing more queries than the active approaches [32, 41, 42]. Papers which actively control the query samples for linear elicitation, e.g. [43], exploit the query space like us in order to achieve lower query complexity. However, unlike us, [43] does not provide theoretical bounds and is also applied to a different query space.

4.9 CONCLUDING REMARKS

We study the space of multiclass confusions and propose efficient algorithms to elicit diagonal-linear and linear performance metrics. We theoretically show that the procedures are robust under feedback and finite sample noise and validate the latter empirically via simulated oracles. We extend elicitation to other families e.g. linear-fractional metrics, thus covering a wide range of metrics encountered in practice.

CHAPTER 5: FAIR PERFORMANCE METRIC ELICITATION

Machine learning models are increasingly employed for critical decision-making tasks such as hiring and sentencing [10, 11, 44, 45, 46]. Yet, it is increasingly evident that automated decision-making is susceptible to bias, whereby decisions made by the algorithm are unfair to certain subgroups [44, 46, 47, 48, 49]. To this end, a wide variety of group fairness metrics have been proposed – all to reduce discrimination and bias from automated decision-making [9, 13, 16, 50, 51, 52]. However, a dearth of formal principles for selecting the most appropriate metric has highlighted the confusion of experts, practitioners, and end users in deciding which group fairness metric to employ [22]. This is further exacerbated by the observation that common metrics often lead to contradictory outcomes [13].

While the problem of selecting an appropriate fairness metric has gained prominence in recent years [16, 22, 52], it perhaps best understood as a special case of the task of choosing evaluation metrics in machine learning. For instance, when a cost-sensitive predictive model classifies patients into cancer categories [53] even without considering fairness, it is often unclear how the cost-tradeoffs be chosen so that they reflect the expert's decision-making, i.e., replacing expert intuition by quantifiable metrics. The proposed Metric Elicitation (ME) framework provides a solution.

Existing research suggests a fundamental trade-off between algorithmic fairness and performance [11, 22, 50, 52, 54, 55], where in addition to appropriate metrics, the practitioner or policymaker must choose a trade-off operating point between the competing objectives [22]. To this end, in this chapter, we extend the ME framework from eliciting multiclass classification metrics to the task of eliciting *fair* performance metrics from pairwise preference feedback in the presence of multiple sensitive groups. In particular, we elicit metrics that reflect, jointly, the (i) predictive performance evaluated as a weighting of classifier's overall predictive rates, (ii) fairness violation assessed as the discrepancy in predictive rates among groups, and (iii) a trade-off between the predictive performance and fairness violation. Importantly, the elicited metrics are sufficiently flexible to encapsulate and generalize many existing predictive performance and fairness violation measures.

In eliciting group-fair performance metrics, we tackle three new challenges. First, from preference query perspective, the predictive performance and fairness violations are correlated, thus increasing the complexity of joint elicitation. Second, we find that in order to measure both positive and negative violations, the fair metrics are necessarily non-linear functions of the predictive rates, thus existing results on linear ME from previous chapters cannot be applied directly. Finally, as we show, the number of groups directly impacts query complexity. We overcome these challenges by proposing a novel query efficient procedure that exploits the geometric properties of the set of predictive rates.

Contributions. We consider metrics for algorithmically group-fair classification and propose a novel approach for eliciting predictive performance, fairness violations, and their trade-off point, from expert pairwise feedback. Our procedure uses binary-search based subroutines and recovers the metric with linear query complexity. Moreover, the procedure is robust to both finite sample and oracle feedback noise thus is useful in practice. Lastly, our method can be applied either by querying preferences over classifiers or predictive rates, which is our choice of measurements (classifier statistics) for this chapter. All the proofs in this chapter are provided in Appendix C.

Notations. Matrices and vectors are denoted by bold upper case and bold lower case letters, respectively. The group membership is denoted by superscripts and coordinates of vectors, matrices, and tuples are denoted by subscripts.

5.1 BACKGROUND

The standard multiclass, multigroup classification setting comprises k classes and m groups with $X \in \mathcal{X}$, $G \in [m]$ and $Y \in [k]$ representing the input, group membership, and output random variables, respectively. The groups are assumed to be disjoint and known apriori [13, 16]. We have access to a dataset $\{(\mathbf{x}, g, y)_i\}_{i=1}^n$ of size n, generated *iid* from a distribution $\mathbb{P}(X, G, Y)$. The measurements (classifier statistics) that we choose to work with in this chapter are the group-specific rates and the overall rates, which are described below.

Group-specific rates: We consider separate (randomized) classifiers $h^g : \mathcal{X} \rightarrow \Delta_k$ for each group g, and use

$$\mathcal{H}^g = \{h^g : \mathcal{X} \rightarrow \Delta_k\} \tag{5.1}$$

to denote the set of all classifiers for group g. The group-specific rate matrix $\mathbf{R}^g(h^g, \mathbb{P}) \in \mathbb{R}^{k \times k}$ for a classifier h^g is given by:

$$R^g_{ij}(h^g, \mathbb{P}) := \mathbb{P}(h^g = j | Y = i, G = g) \quad \text{for } i, j \in [k]. \tag{5.2}$$

Notice that the predictive rates satisfy the following useful decomposition:

$$R^g_{ii}(h^g, \mathbb{P}) = 1 - \sum\nolimits_{j=1, j \neq i}^{k} R^g_{ij}(h^g, \mathbb{P}), \tag{5.3}$$

any rate matrix is uniquely represented by its $q := (k^2 - k)$ off-diagonal elements as a vector $\mathbf{r}^g(h^g, \mathbb{P}) = \textit{off-diag}(\mathbf{R}^g(h^g, \mathbb{P}))$. So we will interchangeably refer to the rate matrix as a *'vector of rates'*. The feasible set of rates associated with a group g is denoted by

$$\mathcal{R}^g = \{\mathbf{r}^g(h^g, \mathbb{P}) \, : \, h^g \in \mathcal{H}^g\}. \tag{5.4}$$

For clarity, we will suppress the dependence on $\mathbb{P}$ and h^g if it is clear from the context.

Overall rates: We define the overall classifier $h : (\mathcal{X}, [m]) \to \Delta_k$ by

$$h(\mathbf{x}, g) \coloneqq h^g(\mathbf{x}) \tag{5.5}$$

and denote its tuple of group-specific rates by:

$$\mathbf{r}^{1:m} \coloneqq (\mathbf{r}^1, \dots, \mathbf{r}^m) \in \mathcal{R}^1 \times \cdots \times \mathcal{R}^m =: \mathcal{R}^{1:m}. \tag{5.6}$$

This tuple allows us to measure the fairness violation across groups. The fairness violation is believed to be in trade-off with the predictive performance [50, 52, 55]. The latter is measured using the overall rate matrix of the classifier h:

$$R_{ij} \coloneqq \mathbb{P}(h = j|Y = i) = \sum\nolimits_{g=1}^{m} t_i^g R_{ij}^g, \tag{5.7}$$

where $t_i^g \coloneqq \mathbb{P}(G = g|Y = i)$ is the prevalence of group g within class i. For an overall classifier h, the *'vector of rates'* $\mathbf{r} = \textit{off-diag}(\mathbf{R})$ can be conveniently written in terms of its group-specific tuple of rates as

$$\mathbf{r} = \sum_{g=1}^{m} \boldsymbol{\tau}^g \odot \mathbf{r}^g, \tag{5.8}$$

where $\boldsymbol{\tau}^g \coloneqq \textit{off-diag}([\mathbf{t}^g \, \mathbf{t}^g \dots \mathbf{t}^g])$.

Fairness violation measure: The (approximate) fairness of a classifier is often determined by the 'discrepancy' in rates across different groups e.g. *equalized odds* [12, 16]. So given two groups $u, v \in [m]$, we define the discrepancy in their rates as:

$$\mathbf{d}^{uv} \coloneqq |\mathbf{r}^u - \mathbf{r}^v|. \tag{5.9}$$

Since there are m groups, the number of *discrepancy vectors* are $\binom{m}{2}$.

5.1.1 Fair Performance Metric

We aim to elicit a general class of metrics, which recovers and generalizes existing fairness measures, based on trade-off between predictive performance and fairness violation [16, 48, 50, 52, 55]. Let $\phi : [0, 1]^q \to \mathbb{R}$ be the cost of overall misclassification (aka. predictive performance) and $\varphi : [0,1]^{m\times q} \to \mathbb{R}$ be the fairness violation cost for a classifier h determined by the overall rates $\mathbf{r}(h)$ and group discrepancies $\{\mathbf{d}^{uv}(h)\}_{u,v=1,v>u}^{m}$, respectively. Without loss of generality (w.l.o.g.), we assume the metrics ϕ and φ are costs. Moreover, the metrics are scale invariant as global scale does not affect the learning problem [18]; hence let $\phi : [0,1]^q \to [0,1]$ and $\varphi : [0,1]^{m\times q} \to [0,1]$.

Definition 5.1 (Fair Performance Metric)**.** Let ϕ and φ be monotonically increasing linear functions of overall rates and group discrepancies, respectively. The fair metric Ψ is a trade-off between ϕ and φ. In particular, given $\mathbf{a} \in \mathbb{R}^q, \mathbf{a} \geq 0$ (misclassification weights), a set of vectors $\mathbf{B} := \{\mathbf{b}^{uv} \in \mathbb{R}^q, \mathbf{b}^{uv} \geq 0\}_{u,v=1,v>u}^{m}$ (fairness violation weights), and a scalar λ (trade-off) with

$$\|\mathbf{a}\|_2 = 1, \qquad \sum\nolimits_{u,v=1,v>u}^{m} \|\mathbf{b}^{uv}\|_2 = 1, \qquad 0 \leq \lambda \leq 1, \tag{5.10}$$

(w.l.o.g., due to scale invariance), we define the metric Ψ as:

$$\Psi(\mathbf{r}^{1:m}\,;\,\mathbf{a},\mathbf{B},\lambda) \,:=\, \underbrace{(1-\lambda)}_{\text{trade-off}}\underbrace{\langle \mathbf{a},\mathbf{r}\rangle}_{\phi(\mathbf{r})} + \lambda\underbrace{\left(\sum\nolimits_{u,v=1,v>u}^{m}\langle \mathbf{b}^{uv},\mathbf{d}^{uv}\rangle\right)}_{\varphi(\mathbf{r}^{1:m})}. \tag{5.11}$$

Examples of the misclassification cost $\phi(\mathbf{r})$ include cost-sensitive linear metrics [35]. Many existing fairness metrics for two classes and two groups such as *equal opportunity* [16], *balance for the negative class* [13] *error-rate balance* (i.e., $0.5|r_1^1 - r_1^2| + 0.5|r_2^1 - r_2^2|$) [48], *weighted equalized odds* (i.e., $b_1|r_1^1 - r_1^2| + b_2|r_2^1 - r_2^2|$) [16, 55], etc. correspond to fairness violations of the form $\varphi(\mathbf{r}^{1:m})$ considered above. The combination of $\phi(\mathbf{r})$ and $\varphi(\mathbf{r}^{1:m})$ as defined in $\Psi(\mathbf{r}^{1:m})$ appears regularly in prior work [50, 52, 55]. Notice that the metric is flexible to allow different fairness violation costs for different pairs of groups thus capable of enabling reverse discrimination [56]. Lastly, while the metric is linear with respect to (w.r.t.) the discrepancies, it is non-linear w.r.t. the group-wise rates. Hence, standard linear ME algorithm from Chapters 3 and 4 cannot be trivially applied for eliciting the metric in Definition 5.1.

5.1.2 Fair Performance Metric Elicitation; Problem Statement

We now state the problem of *Fair Performance Metric Elicitation (FPME)* and define the associated *oracle query*. The broad definitions follow from Chapter 2, extended so the predictive rates (classifier statistics) and the performance metrics correspond to the multiclass multigroup-fair classification setting.

Definition 5.2 (Oracle Query)**.** Given two classifiers h_1, h_2 (equivalent to a tuple of rates $\mathbf{r}_1^{1:m}, \mathbf{r}_2^{1:m}$ respectively), a query to the Oracle (with metric Ψ) is represented by:

$$\Gamma(h_1, h_2\,;\,\Psi) = \Omega\left(\mathbf{r}_1^{1:m}, \mathbf{r}_2^{1:m}\,;\,\Psi\right) = \mathbf{1}[\Psi(\mathbf{r}_1^{1:m}) > \Psi(\mathbf{r}_2^{1:m})], \tag{5.12}$$

where $\Gamma : \mathcal{H} \times \mathcal{H} \to \{0, 1\}$ and $\Omega : \mathcal{R}^{1:m} \times \mathcal{R}^{1:m} \to \{0, 1\}$. In simple words, the query asks whether h_1 is preferred to h_2 (equivalent to whether $\mathbf{r}_1^{1:m}$ is preferred to $\mathbf{r}_2^{1:m}$), as measured by Ψ.

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 to them via interpretable learning techniques [19, 20] or via A/B testing [21]. For example, one may perform A/B testing for an internet-based application by deploying two classifiers A and B and use the population's level of engagement to decide the preference between the two classifiers. For other applications, intuitive visualizations of the predictive rates for two different classifiers (see e.g., [22, 23]) can be used to ask preference feedback from a group of domain experts.

We emphasize that the metric Ψ used by the oracle is unknown to us and can be accessed only through queries to the oracle. Since the metrics we consider are functions of rates, comparing two classifiers on a metric is equivalent to comparing their corresponding rates. Henceforth, we will denote any query to the oracle by a pair of rates $(\mathbf{r}_1^{1:m}, \mathbf{r}_2^{1:m})$. Also, whenever we refer to an oracles's dimension, we are referring to the dimension of its rate arguments. For instance, we will consider the oracle in Definition 5.2 to be of dimension $m \times q$. Next, we formally state the FPME problem.

Definition 5.3 (Fair Performance Metric Elicitation with Pairwise Comparison Queries (given $\{(\mathbf{x}, g, y)_i\}_{i=1}^n$))**.** Suppose that the oracle's (unknown) performance metric is Ψ. Using oracle queries of the form $\Omega(\hat{\mathbf{r}}_1^{1:m}, \hat{\mathbf{r}}_2^{1:m})$, where $\hat{\mathbf{r}}_1^{1:m}, \hat{\mathbf{r}}_2^{1:m}$ are the estimated rates from samples, recover a metric $\hat{\Psi}$ such that $\|\Psi - \hat{\Psi}\| < \omega$ under a suitable norm $\|\cdot\|$ for sufficiently small error tolerance $\omega > 0$.

Similar to the standard metric elicitation problems (Chapters 3 and 4), the performance of FPME is evaluated both by the fidelity of the recovered metric and the query complexity. As done in decision theory literature [6, 24], we present our FPME solution by first assuming access to population quantities such as the population rates $\mathbf{r}^{1:m}(h, \mathbb{P})$, and then discuss how elicitation can be performed from finite samples, e.g., with empirical rates $\hat{\mathbf{r}}^{1:m}(h, \{(\mathbf{x}, g, y)_i\}_{i=1}^n)$.

5.1.3 Linear Performance Metric Elicitation – Warmup

We revisit the Linear Performance Metric Elicitation (LPME) procedure from Chapter 4, which we will use as as a subroutine to elicit fair performance metrics. The LPME procedure assumes an enclosed sphere $\mathcal{S} \subset \mathcal{Z}$, where $\mathcal{Z}$ is the q-dimensional space of classifier statistics that are feasible, i.e., can be achieved by some classifier. It also assumes access to a q-dimensional oracle Ω' whose scale invariant linear metric is of the form $\xi(\mathbf{z}) := \langle \mathbf{a}, \mathbf{z} \rangle$ with $\|\mathbf{a}\|_2 = 1$, analogous to the misclassification cost in Definition 5.1. Analogously, the oracle queries are of the type $\Omega'(\mathbf{z}_1, \mathbf{z}_2) := \mathbf{1}[\xi(\mathbf{z}_1) > \xi(\mathbf{z}_2)]$.

When the number of classes $k = 2$, LPME elicits the coefficients $\mathbf{a}$ using a simple one-dimensional binary search. When $k > 2$, LPME performs binary search in each coordinate while keeping the others fixed, and performs this in a coordinate-wise fashion until convergence. By restricting this coordinate-wise binary search procedure to posing queries from within a sphere $\mathcal{S}$, LPME can be equivalently seen as minimizing a strongly-convex function and shown to converge to a solution $\hat{\mathbf{a}}$ close to $\mathbf{a}$. Specifically, the algorithm takes the query space $\mathcal{S} \subset \mathcal{Z}$, binary-search tolerance ϵ, and the oracle Ω' as input, and by querying $O(q \log(1/\epsilon))$ queries recovers $\hat{\mathbf{a}}$ with $\|\hat{\mathbf{a}}\|_2 = 1$ such that $\|\mathbf{a} - \hat{\mathbf{a}}\|_2 \leq O(\sqrt{q}\epsilon)$ (Theorem 4.2 in Chapter 4). Please see the details of the LPME procedure in Algorithm 4.2 (Chapter 4) for completeness. We summarize the discussion with the following remark.

Remark 5.1. Given a q-dimensional space $\mathcal{Z}$ enclosing a sphere $\mathcal{S} \subset \mathcal{Z}$ and an oracle Ω' with linear metric $\xi(\mathbf{z}) := \langle \mathbf{a}, \mathbf{z} \rangle$, the LPME algorithm (Algorithm 4.2, Chapter 4) 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., ${}^{a_i}/_{a_j} \approx {}^{\hat{a}_i}/_{\hat{a}_j} \; \forall \, i, j \in [q]$.

Note that the algorithm estimates the direction of the coefficient vector, not its magnitude.

5.2 GEOMETRY OF THE PRODUCT SET $\mathcal{R}^{1:M}$

The LPME procedure described above works with rate queries of dimension q. We would like to use this procedure to elicit the fair metrics in Definition 5.1 defined on tuples of dimension $m \times q$. So to make use of LPME, we restrict our queries to a q-dimensional sphere $\mathcal{S}$ which is common to the feasible rate region $\mathcal{R}^g$ for each group g, i.e., to a sphere in the intersection $\mathcal{R}^1 \cap \ldots \cap \mathcal{R}^m$. We show now that such a sphere does indeed exist under a mild assumption.

Assumption 5.1. For all groups, the conditional-class distributions are not identical, i.e., $\forall \, g \in [m], \forall \, i \neq j$, $\mathbb{P}(Y = i|X, G = g) \neq \mathbb{P}(Y = j|X, G = g)$. In other words, there is some non-trivial signal for classification for each group.

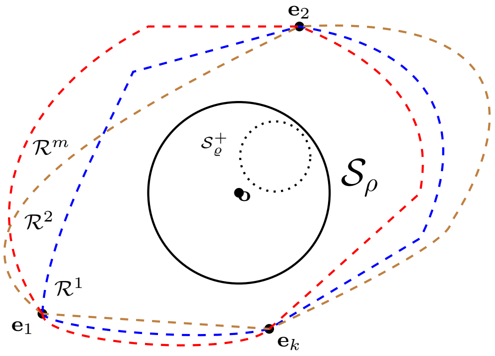


Figure 5.1: $\mathcal{R}^1 \times \cdots \times \mathcal{R}^m$ (best seen in colors); $\mathcal{R}^u \, \forall \, u \in [m]$ are convex sets with common vertices $\mathbf{e}_i \, \forall \, i \in [k]$ and enclose the sphere $\mathcal{S}_\rho$.

Let $\mathbf{e}_i \in \{0, 1\}^q$ be the rate profile for a trivial classifier that predicts class i on all inputs. Note that these trivial classifiers evaluate to the same rates $\mathbf{e}_i$ irrespective of which group we apply them to.

Proposition 5.1 (Geometry of $\mathcal{R}^{1:m}$; Figure 5.1)**.** For any group $g \in [m]$, the set of confusion rates $\mathcal{R}^g$ is convex, bounded in $[0, 1]^q$, and has vertices $\{\mathbf{e}_i\}_{i=1}^k$. The intersection of group rate sets $\mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$ is convex and always contains the rate $\mathbf{o} = \frac{1}{k}\sum_{i=1}^k \mathbf{e}_i$ in the interior, which is associated with the uniform random classifier that predicts each class with equal probability.

Since $\mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$ is convex and always contains a point $\mathbf{o}$ in the interior, we can make the following remark (see Figure 5.1 for an illustration).

Remark 5.2 (Existence of common sphere $\mathcal{S}_\rho$)**.** There exists a q-dimensional sphere $\mathcal{S}_\rho \subset \mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$ of non-zero radius ρ centered at $\mathbf{o}$. Thus, any rate $\mathbf{s} \in \mathcal{S}_\rho$ is feasible for all groups, i.e., $\mathbf{s}$ is achievable by some classifier h^g for all groups $g \in [m]$.

A method to obtain $\mathcal{S}_\rho$ with suitable radius ρ from Chapter 4 is discussed in Appendix C.1.1. From Remark 5.2, we observe that any tuple of group rates $\mathbf{r}^{1:m} = (\mathbf{s}^1, \ldots, \mathbf{s}^m)$ chosen from $\mathcal{S}_\rho \times \ldots \times \mathcal{S}_\rho$ is achievable for some choice of group-specific classifiers $h^1, \ldots, h^m$. Moreover, when two groups u, v are assigned the same rate profile $\mathbf{s} \in \mathcal{S}_\rho$, the fairness discrepancy $\mathbf{d}^{uv} = \mathbf{0}$. We will exploit these observations in the elicitation strategy we discuss next.

5.3 METRIC ELICITATION

We have access to an oracle whose (unknown) metric $\overline{\Psi}$ given in Definition 5.1 is parameterized by $(\bar{\mathbf{a}}, \overline{\mathbf{B}}, \bar{\lambda})$. The proposed FPME framework for eliciting the oracle's metric

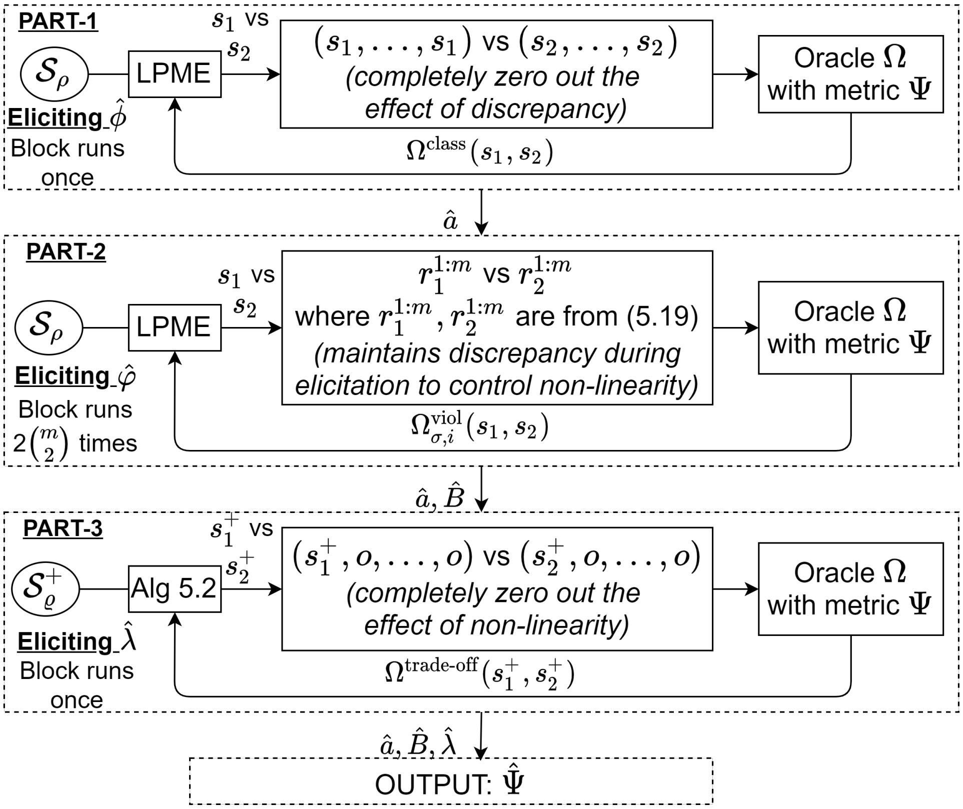

PART-1
$\mathcal{S}_\rho$
LPME
s_1 vs s_2
$(s_1, \dots, s_1)$ vs $(s_2, \dots, s_2)$
(completely zero out the effect of discrepancy)
Oracle Ω with metric Ψ
$\Omega^{\text{class}}(s_1, s_2)$
Eliciting $\hat{\phi}$
Block runs once
$\hat{a}$
PART-2
$\mathcal{S}_\rho$
LPME
s_1 vs s_2
$r_1^{1:m}$ vs $r_2^{1:m}$
where $r_1^{1:m}, r_2^{1:m}$ are from (5.19)
(maintains discrepancy during elicitation to control non-linearity)
Oracle Ω with metric Ψ
$\Omega^{\text{viol}}_{\sigma,i}(s_1, s_2)$
Eliciting $\hat{\varphi}$
Block runs $2\binom{m}{2}$ times
$\hat{a}, \hat{B}$
PART-3
$\mathcal{S}^+_\varrho$
Alg 5.2
s_1^+ vs s_2^+
$(s_1^+, o, \dots, o)$ vs $(s_2^+, o, \dots, o)$
(completely zero out the effect of non-linearity)
Oracle Ω with metric Ψ
$\Omega^{\text{trade-off}}(s_1^+, s_2^+)$
Eliciting $\hat{\lambda}$
Block runs once
$\hat{a}, \hat{B}, \hat{\lambda}$
OUTPUT: $\hat{\Psi}$

Figure 5.2: Workflow of the FPME procedure.

is presented in Figure 5.2 and is summarized in Algorithm 5.1. The procedure has three parts executed in sequence: (a) eliciting the misclassification cost $\overline{\phi}(\mathbf{r})$ (i.e., $\overline{\mathbf{a}}$), (b) eliciting the fairness violation $\overline{\varphi}(\mathbf{r}^{1:m})$ (i.e., $\overline{\mathbf{B}}$), and (c) eliciting the trade-off between the misclassification cost and fairness violation (i.e., $\overline{\lambda}$). For simplicity, we will suppress the coefficients $(\overline{\mathbf{a}}, \overline{\mathbf{B}}, \overline{\lambda})$ from the notation Ψ whenever it is clear from context.

Notice that the metric Ψ is *piece-wise linear* in its coefficients. So our high level idea is to restrict the queries we pose to the oracle to lie within regions where the metric Ψ is linear, so that we can then employ the LPME subroutine to elicit the corresponding linear coefficients. We will show for each of the three components (a)–(c), how we can identify regions in the query space where the metric is linear and apply the LPME procedure (or a variant of it). By restricting the query inputs to those regions, we will essentially be converting the $(m \times q)$-dimensional oracle Ω in Definition 5.2 into an equivalent q-dimensional oracle that compares rates $\mathbf{s}_1, \mathbf{s}_2$ from the common sphere $\mathcal{S}_\rho \subset \mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$. We first discuss our approach assuming the oracle has no *feedback* noise, and later in Section 5.4 show that our approach is robust to noisy feedback and provide query complexity guarantees.

Algorithm 5.1 FPM Elicitation

```
 1: **Input:** Query spaces $\mathcal{S}_\rho$, $\mathcal{S}_\varrho^+$, search tolerance $\epsilon > 0$, and oracle $\Omega$
 2: $\hat{\mathbf{a}} \leftarrow$ LPME$(\mathcal{S}_\rho, \epsilon, \Omega^{\text{class}})$
 3: **if** $m == 2$ **then**
 4:     $\breve{\mathbf{f}} \leftarrow$LPME$(\mathcal{S}_\rho, \epsilon, \Omega_1^{\text{viol}})$
 5:     $\tilde{\mathbf{f}} \leftarrow$LPME$(\mathcal{S}_\rho, \epsilon, \Omega_2^{\text{viol}})$
 6:     $\hat{\mathbf{b}}^{12} \leftarrow$ normalized solution from (5.18)
 7: **else**
 8:     Let $\mathcal{L} \leftarrow \varnothing$
 9:     **for** $\sigma \in \mathcal{M}$ **do**
10:         $\breve{\mathbf{f}}^\sigma \leftarrow$LPME$(\mathcal{S}_\rho, \epsilon, \Omega_{\sigma,1}^{\text{viol}})$
11:         $\tilde{\mathbf{f}}^\sigma \leftarrow$LPME$(\mathcal{S}_\rho, \epsilon, \Omega_{\sigma,k}^{\text{viol}})$
12:         Let $\ell^\sigma$ be Eq. (5.20), extend $\mathcal{L} \leftarrow \mathcal{L} \cup \{\ell^\sigma\}$
13:     **end for**
14:     $\hat{\mathbf{B}} \leftarrow$ normalized solution from (5.21) using $\mathcal{L}$
15: **end if**
16: $\hat{\lambda} \leftarrow$ Algorithm 5.2 $(\mathcal{S}_\varrho^+, \epsilon, \Omega^{\text{trade-off}})$
17: **Output:** $\hat{\mathbf{a}}, \hat{\mathbf{B}}, \hat{\lambda}$
```

5.3.1 Eliciting the Misclassification Cost $\overline{\phi}(\mathbf{r})$: Part 1 in Figure 5.2 and Line 1 in Algorithm 5.1

To elicit the misclassification cost coefficients $\bar{\mathbf{a}}$, we will query from a region of the query space where the fairness violation term in the metric is zero. Specifically, we will query group rate profile of the form $\mathbf{r}^{1:m} = (\mathbf{s}, \ldots, \mathbf{s})$, where $\mathbf{s}$ is a q-dimensional rate from the common sphere $\mathcal{S}_\rho$. For these group rate profiles, the metric Ψ simply evaluates to the linear misclassification term, i.e.:

$$\overline{\Psi}(\mathbf{s}, \ldots, \mathbf{s}) = (1 - \overline{\lambda})\langle \bar{\mathbf{a}}, \mathbf{s} \rangle. \tag{5.13}$$

So given a pair of group rate profiles $\mathbf{r}_1^{1:m} = (\mathbf{s}_1, \ldots, \mathbf{s}_1)$ and $\mathbf{r}_2^{1:m} = (\mathbf{s}_2, \ldots, \mathbf{s}_2)$, where $\mathbf{s}_1, \mathbf{s}_2 \in \mathcal{S}_\rho$, the oracle's response will essentially compare $\mathbf{s}_1$ and $\mathbf{s}_2$ on the linear metric $(1-\overline{\lambda})\langle \bar{\mathbf{a}}, \mathbf{s} \rangle$. Hence, we estimate the coefficients $\bar{\mathbf{a}}$ by applying LPME over the q-dimensional sphere $\mathcal{S}_\rho$ with a modified oracle Ω^{class} which takes a pair of rate profiles $\mathbf{s}_1$ and $\mathbf{s}_2$ from $\mathcal{S}_\rho$ as input, and responds with:

$$\Omega^{\text{class}}(\mathbf{s}_1, \mathbf{s}_2) \;=\; \Omega((\mathbf{s}_1, \ldots, \mathbf{s}_1),\, (\mathbf{s}_2, \ldots, \mathbf{s}_2)). \tag{5.14}$$

This is decribed in line 1 of Algorithm 5.1, which applies the LPME subroutine with query space $\mathcal{S}_\rho$, binary search tolerance ϵ, and the oracle Ω^{class}. From Remark 5.1, this subroutine returns a coefficient vector $\mathbf{f}$ with $\|\mathbf{f}\|_2 = 1$ such that:

$$\frac{(1-\bar{\lambda})a_i}{(1-\bar{\lambda})a_j} = \frac{f_i}{f_j} \implies \frac{a_i}{a_j} = \frac{f_i}{f_j}. \tag{5.15}$$

By setting $\hat{\mathbf{a}} = \mathbf{f}$, we recover the classification coefficients independent of the fairness violation coefficients and trade-off parameter. See part 1 in Figure 5.2 for further illustration.

5.3.2 Eliciting the Fairness Violation $\overline{\varphi}(\mathbf{r}^{1:m})$: Part 2 in Figure 5.2 and lines 3-15 in Algorithm 5.1

We now discuss eliciting the fairness term $\overline{\varphi}(\mathbf{r}^{1:m})$. We will first discuss the special case of $m = 2$ groups and later discuss how the proposed procedure can be extended to handle multiple groups.

Special Case of $m = 2$: Lines 4-6 in Algorithm 5.1: Recall from Definition 5.1 that in the violation term, we measure the group discrepancies using the *absolute* difference between the group rates, i.e., $\mathbf{d}^{12} = |\mathbf{r}^1 - \mathbf{r}^2|$. If we restrict our queries to only those rate profiles $\mathbf{r}^{1:2}$ for which the difference in each coordinate of $\mathbf{r}^1 - \mathbf{r}^2$ is either always positive or always negative, then we can treat the violation term as a linear metric within this region and apply LPME to estimate the associated coefficients.

To this end, we pose to the oracle queries of the form $\mathbf{r}^{1:2} = (\mathbf{s}, \mathbf{e}_i)$, where we assign to group 1 a rate profile $\mathbf{s}$ from the common sphere $\mathcal{S}_\rho$, and to group 2 the rate profile $\mathbf{e}_i \in \{0,1\}^q$ for some i. Remember that $\mathbf{e}_i$ is a rate vector associated with a trivial classifier which predicts class i on all inputs, and is therefore a binary vector. Since we know whether an entry of $\mathbf{e}_i$ is either a 0 or a 1, we can decipher the signs of each entry of the difference vector $\mathbf{s} - \mathbf{e}_i$. Hence for group rate profiles of the above form, the metric Ψ can be written as a linear function in $\mathbf{s}$:

$$\overline{\Psi}(\mathbf{s}, \mathbf{e}_i) = \langle (1-\bar{\lambda})\bar{\mathbf{a}} \odot (\mathbf{1} - \boldsymbol{\tau}^2) + \bar{\lambda}\mathbf{w}_i \odot \bar{\mathbf{b}}^{12}, \mathbf{s}\rangle + c_i, \tag{5.16}$$

where $\mathbf{w}_i := 1 - 2\mathbf{e}_i$ tells us the sign of each entry of $\mathbf{s} - \mathbf{e}_i$, c_i is a constant, and we have used the fact that $\boldsymbol{\tau}^1 = \mathbf{1} - \boldsymbol{\tau}^2$. Fixing a class i, we then apply LPME over the q-dimensional sphere $\mathcal{S}_\rho$ with a modified oracle Ω_i^{viol} which takes a pair of rate profiles $\mathbf{s}_1, \mathbf{s}_2 \in \mathcal{S}_\rho$ as input and responds with:

$$\Omega_i^{\text{viol}}(\mathbf{s}_1, \mathbf{s}_2) \;=\; \Omega((\mathbf{s}_1, \mathbf{e}_i), (\mathbf{s}_2, \mathbf{e}_i)). \tag{5.17}$$

One run of LPME with oracle Ω_1^{viol} results in $q-1$ independent equations. In order to elicit a q-dimensional vector $\mathbf{b}^{12}$, we must run LPME again with oracle Ω_2^{viol}. This is described in lines 4 and 5 of Algorithm 5.1. The LPME calls provide us with two slopes $\breve{\mathbf{f}}, \tilde{\mathbf{f}}$ such that $\|\breve{\mathbf{f}}\|_2 = \|\tilde{\mathbf{f}}\|_2 = 1$ from which it is easy to obtain the fairness violation weights:

$$\hat{\mathbf{b}}^{12} = \frac{\tilde{\mathbf{b}}^{12}}{\|\tilde{\mathbf{b}}^{12}\|_2}, \quad \text{with} \quad \tilde{\mathbf{b}}^{12} = \mathbf{w}_1 \odot \left[\delta\breve{\mathbf{f}} - \hat{\mathbf{a}} \odot (\mathbf{1} - \boldsymbol{\tau}^2)\right], \tag{5.18}$$

where δ is a scalar depending on the known entities $\boldsymbol{\tau}^{12}, \hat{\mathbf{a}}, \breve{\mathbf{f}}^{12}, \tilde{\mathbf{f}}^{12}$. The derivation is provided in Appendix C.2.2 for completeness. Because $\overline{\varphi}$ is scale invariant (see Definition 5.1), the normalized solution $\hat{\mathbf{b}}^{12}$ is independent of the true trade-off $\overline{\lambda}$ and depends only on the previously elicited vector $\hat{\mathbf{a}}$.

General Case of $m > 2$: Lines 8-14 in Algorithm 5.1: We briefly outline the elicitation procedure for $m > 2$ groups, with details in Appendix C.2.2. Let $\mathcal{M}$ be a set of subsets of the m groups such that each element $\sigma \in \mathcal{M}$ and $[m] \setminus \sigma$ partition the set of m groups. We will later discuss how to choose $\mathcal{M}$ for efficient elicitation. Similar to the two-group case, we pose queries $\mathbf{r}^{1:m}$ where to a subset of groups $\sigma \in \mathcal{M}$, we assign the trivial rate vector $\mathbf{e}_i$ and to the rest $[m] \setminus \sigma$ groups, we assign a point $\mathbf{s}$ from the common sphere $\mathcal{S}_\rho$. Observe that within this query region, the metric Ψ is linear in its inputs. So for a fixed partitioning of groups defined by σ, we apply LPME with a query space $\mathcal{S}_\rho$ using the modified q-dimensional oracle:

$$\Omega_{\sigma,i}^{\text{viol}}(\mathbf{s}_1, \mathbf{s}_2) = \Omega(\mathbf{r}_1^{1:m}, \mathbf{r}_2^{1:m}) \;\; \text{where} \;\; \mathbf{r}_1^g = \begin{cases} \mathbf{e}_i & \text{if } g \in \sigma \\ \mathbf{s}_1 & \text{o.w.} \end{cases} \;\; \text{and} \;\; \mathbf{r}_2^g = \begin{cases} \mathbf{e}_i & \text{if } g \in \sigma \\ \mathbf{s}_2 & \text{o.w.} \end{cases}. \tag{5.19}$$

As described in lines 10 and 11 of the algorithm, we repeat this twice fixing class i to 1 and k. The guarantees for LPME then give us the following relationship between coefficients $\overline{\mathbf{b}}^{uv}$ we wish to elicit and the already elicited coefficient $\hat{\mathbf{a}}$:

$$\sum\nolimits_{u,v} \mathbf{1}\left[|\{u, v\} \cap \sigma| = 1\right] \tilde{\mathbf{b}}^{uv} = \mathbf{w}_1 \odot \left[\delta^\sigma \breve{\mathbf{f}}^\sigma - \hat{\mathbf{a}} \odot (\mathbf{1} - \boldsymbol{\tau}^\sigma)\right], \tag{5.20}$$

where $\boldsymbol{\tau}^\sigma = \sum_{g\in\sigma} \boldsymbol{\tau}^g$ and $\tilde{\mathbf{b}}^{uv} := \overline{\lambda}\overline{\mathbf{b}}^{uv}/(1 - \overline{\lambda})$ is a scaled version of the true (unknown) $\overline{\mathbf{b}}^{uv}$. Since we need to estimate $\binom{m}{2}$ coefficients, we repeat the above procedure for $\binom{m}{2}$ partitions of the groups defined by σ and get a system of $\binom{m}{2}$ linear equations. We may choose any $\mathcal{M}$ of size $\binom{m}{2}$ so that the equations are independent. From the solution to these equations, we recover $\tilde{\mathbf{b}}^{uv}$'s, which we further normalize to get estimates of the final fairness violation weights:

$$\hat{\mathbf{b}}^{uv} = \frac{\tilde{\mathbf{b}}^{uv}}{\sum_{u,v=1,v>u}^{m} \|\tilde{\mathbf{b}}^{uv}\|_2} \quad \text{for} \quad u, v \in [m], v > u. \tag{5.21}$$

Because of normalization, the elicited fairness weights are independent of the trade-off $\bar{\lambda}$.

5.3.3 Eliciting Trade-off $\bar{\lambda}$: Part 3 in Figure 5.2 and Line 16 in Algorithm 5.1

Equipped with estimates of the misclassification and fairness violation coefficients $(\hat{\bar{\mathbf{a}}}, \hat{\bar{\mathbf{B}}})$, the final step is to elicit the trade-off $\bar{\lambda}$ between them. We now show how this can be posed as one-dimensional binary search problem. Suppose we restrict our queries to be of the form $\mathbf{r}^{1:m} = (\mathbf{s}^+, \mathbf{o}, \dots, \mathbf{o})$, where for all but the first group, we assign the rate $\mathbf{o}$ associated with a uniform random classifier, and for the first group, we assign some rate $\mathbf{s}^+$ such that $\mathbf{s}^+ \geq \mathbf{o}$. For these rate profiles, the group rate difference terms $\mathbf{r}^1 - \mathbf{r}^v = \mathbf{s}^+ - \mathbf{o} \geq \mathbf{0}$ for all $v \in \{2, \dots, m\}$, and all the other difference terms are $\mathbf{0}$. As a result, the metric Ψ is linear in the input rate profiles:

$$\overline{\Psi}(\mathbf{s}^+, \mathbf{o}, \dots, \mathbf{o}) = \langle (1 - \bar{\lambda}) \boldsymbol{\tau}^1 \odot \bar{\mathbf{a}} + \bar{\lambda} \sum\nolimits_{v=2}^{m} \bar{\mathbf{b}}^{1v}, \mathbf{s}^+ \rangle + c, \tag{5.22}$$

where c is a constant. Despite the metric being linear in the identified input region, we cannot directly apply the LPME procedure described in Section 5.1.3 to elicit λ, because we have one parameter to elicit but the input to the metric is q-dimensional. Here we propose a slight variant of LPME.

Similar to the original ME procedure for the binary classification setup in Chapter 3, we first construct a one-dimensional function ϑ, which takes a guess of the trade-off parameter as input, and outputs the quality of the guess. We show that this function is unimodal and its mode coincides with the oracle's true trade-off parameter λ.

Lemma 5.1. Let $\mathcal{S}_\varrho^+ \subset \mathcal{S}_\rho$ be a q-dimensional sphere with radius $\varrho < \rho$ such that $\mathbf{s}^+ \geq \mathbf{o}$, $\forall\, \mathbf{s}^+ \in \mathcal{S}_\varrho^+$ (see Figure 5.1). Assume the estimates $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}^{uv}$'s satisfy a mild regularity condition $\langle \hat{\mathbf{a}}, \sum_{v=2}^{m} \hat{\mathbf{b}}^{1v} \rangle \neq 1$. Define a one-dimensional function ϑ as:

$$\vartheta(\bar{\lambda}) := \Psi(\mathbf{s}_{\bar{\lambda}}^*, \mathbf{o}, \dots, \mathbf{o}), \tag{5.23}$$

where

$$\mathbf{s}_{\bar{\lambda}}^* \;=\; \underset{s^+ \in \mathcal{S}_\varrho^+}{\operatorname{argmax}} \langle (1 - \bar{\lambda}) \boldsymbol{\tau}^1 \odot \hat{\bar{\mathbf{a}}} + \bar{\lambda} \sum\nolimits_{v=2}^{m} \hat{\bar{\mathbf{b}}}^{1v}, \mathbf{s}^+ \rangle. \tag{5.24}$$

Then the function ϑ is strictly quasiconcave (and therefore unimodal) in $\bar{\lambda}$. Moreover, the

Algorithm 5.2 Eliciting the trade-off $\overline{\lambda}$

```
**Input:** Query space $\mathcal{S}_\varrho^+$, binary-search tolerance $\epsilon > 0$, oracle $\Omega^{\text{trade-off}}$
**Initialize:** $\lambda^{(a)} = 0$, $\lambda^{(b)} = 1$.
**while** $\left|\lambda^{(b)} - \lambda^{(a)}\right| > \epsilon$ **do**
    Set $\lambda^{(c)} = \frac{3\lambda^{(a)}+\lambda^{(b)}}{4}$, $\lambda^{(d)} = \frac{\lambda^{(a)}+\lambda^{(b)}}{2}$, $\lambda^{(e)} = \frac{\lambda^{(a)}+3\lambda^{(b)}}{4}$
    Set $\mathbf{s}^{(a)} = \operatorname*{argmax}_{\mathbf{s}^+ \in \mathcal{S}_\varrho^+} \langle (1-\lambda_a)\boldsymbol{\tau}^1 \odot \hat{\mathbf{a}} + \lambda_a \sum_{v=2}^{m} \hat{\mathbf{b}}^{1v}, \mathbf{s}^+ \rangle$ using Lemma 4.1 (Chapter 4)
    Similarly, set $\mathbf{s}^{(c)}$, $\mathbf{s}^{(d)}$, $\mathbf{s}^{(e)}$, $\mathbf{s}^{(b)}$.
    Query $\Omega^{\text{trade-off}}(\mathbf{s}^{(c)}, \mathbf{s}^{(a)})$, $\Omega^{\text{trade-off}}(\mathbf{s}^{(d)}, \mathbf{s}^{(c)})$, $\Omega^{\text{trade-off}}(\mathbf{s}^{(e)}, \mathbf{s}^{(d)})$, and $\Omega^{\text{trade-off}}(\mathbf{s}^{(b)}, \mathbf{s}^{(e)})$.
    $[\lambda^{(a)}, \lambda^{(b)}] \leftarrow$ *ShrinkInterval* (responses) using a subroutine analogous to the routine shown in Figure B.1.
**end while**
**Output:** $\hat{\lambda} = \frac{\lambda^{(a)}+\lambda^{(b)}}{2}$.
```

mode of this function is achieved at the oracle's true trade-off parameter λ.

For a candidate trade-off $\bar{\lambda}$, the function ϑ first constructs a candidate linear metric based on (5.22), maximizes this candidate metric over inputs $\mathbf{s}^+$, and evaluates the oracle's true metric Ψ at the maximizing rate profile. Note that we cannot directly compute the function ϑ as it needs the oracle's metric Ψ. However, given two candidates for the trade-off parameter $\bar{\lambda}_1$ and $\bar{\lambda}_2$, one can compare the values of $\vartheta(\bar{\lambda}_1)$ and $\vartheta(\bar{\lambda}_2)$ by finding the corresponding maximizers over $\mathbf{s}^+$ and querying the oracle to compare them. Because ϑ is unimodal, one can use a simple binary search using such pairwise comparisons to find the mode of the function, which we know coincides with the true λ.

We provide an outline of this procedure in Algorithm 5.2, which uses the modified oracle

$$\Omega^{\text{trade-off}}(\mathbf{s}_1^+, \mathbf{s}_2^+) = \Omega((\mathbf{s}_1^+, \mathbf{o}, \ldots, \mathbf{o}),\, (\mathbf{s}_2^+, \mathbf{o}, \ldots, \mathbf{o})) \tag{5.25}$$

to compare the maximizers in (5.24).

Description of Algorithm 5.2: Given the unimodality of $\vartheta(\lambda)$ from Lemma 5.1, we devise the binary-search procedure Algorithm 5.2 for eliciting the true trade-off $\bar{\lambda}$. The algorithm takes in input the query space $\mathcal{S}_\varrho^+$, binary-search tolerance ϵ, an equivalent oracle $\Omega^{\text{trade-off}}$, the elicited $\hat{\mathbf{a}}$ from Section 5.3.1, and the elicited $\hat{\mathbf{B}}$ from Section 5.3.2. The algorithm finds the maximizer of the function $\hat{\vartheta}(\lambda)$ defined analogously to (5.23), where $\bar{\mathbf{a}}, \overline{\mathbf{B}}$ are replaced by $\hat{\mathbf{a}}, \hat{\mathbf{B}}$, using Lemma 4.1 (Chapter 4). The algorithm poses four queries to the oracle and shrink the interval $[\lambda^{(a)}, \lambda^{(b)}]$ into half based on the responses using a subroutine analogous to *ShrinkInterval* shown in Figure B.1. The algorithm stops when the length of the search interval $[\lambda^{(a)}, \lambda^{(b)}]$ is less than the tolerance ϵ. Combining parts 1, 2 and 3 in Figure 5.2 completes the FPME procedure.

5.4 GUARANTEES

We discuss elicitation guarantees under the following feedback model.

Definition 5.4 (Oracle Feedback Noise: $\epsilon_\Omega \geq 0$)**.** For two rates $\mathbf{r}_1^{1:m}, \mathbf{r}_2^{1:m} \in \mathcal{R}^{1:m}$, the oracle responds correctly as long as $|\overline{\Psi}(\mathbf{r}_1^{1:m}) - \overline{\Psi}(\mathbf{r}_2^{1:m})| > \epsilon_\Omega$. Otherwise, it may be incorrect.

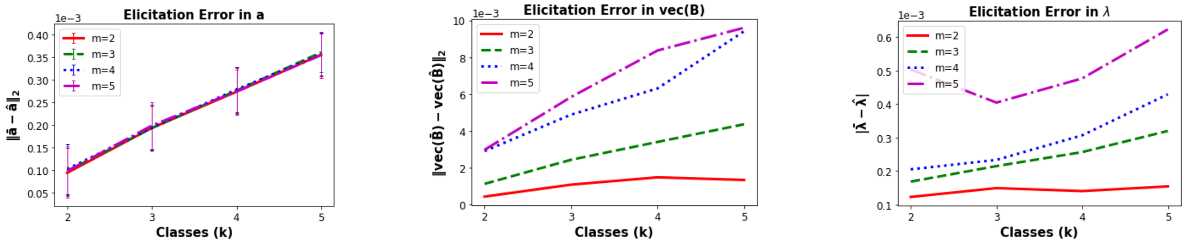
In words, the oracle may respond incorrectly if the rates are very close as measured by the metric $\overline{\Psi}$. Since deriving the final metric involves offline computations including certain ratios, we discuss guarantees under a regularity assumption that ensures all components are well defined.

Assumption 5.2. We assume that $1 > c_1 > \bar{\lambda} > c_2 > 0$, $\min_i |a_i| > c_3$, $\min_i |(1-\bar{\lambda})a_i\tau_i^\sigma - \bar{\lambda}w_{ji}b_i^\sigma| > c_4 \,\forall\, j \in [q], \sigma \in \mathcal{M}$, for some $c_1, c_2, c_3, c_4 > 0$, $\rho > \varrho \gg \epsilon_\Omega$, and $\langle \bar{\mathbf{a}}, \sum_{v=2}^m \bar{\mathbf{b}}^{1v} \rangle \neq 1$.

Theorem 5.1. Given $\epsilon, \epsilon_\Omega \geq 0$, and a 1-Lipschitz fair performance metric $\overline{\Psi}$ parametrized by $\bar{\mathbf{a}}, \overline{\mathbf{B}}, \bar{\lambda}$, under Assumptions 5.1 and 5.2, Algorithm 5.1 returns a metric $\hat{\Psi}$ with parameters:

- $\hat{\mathbf{a}}$: after $O\left(q \log \frac{1}{\epsilon}\right)$ queries such that $\|\bar{\mathbf{a}} - \hat{\mathbf{a}}\|_2 \leq O\left(\sqrt{q}(\epsilon + \sqrt{\epsilon_\Omega/\rho})\right)$.
- $\hat{\mathbf{B}}$: after $O\left(\binom{m}{2} q \log \frac{1}{\epsilon}\right)$ queries such that $\|\text{vec}(\overline{\mathbf{B}}) - \text{vec}(\hat{\mathbf{B}})\|_2 \leq O\left(mq(\epsilon + \sqrt{\epsilon_\Omega/\rho})\right)$, where vec$(\cdot)$ vectorizes the matrix.
- $\hat{\lambda}$: after $O(\log(\frac{1}{\epsilon}))$ queries, with error $|\bar{\lambda} - \hat{\lambda}| \leq O\left(\epsilon + \sqrt{\epsilon_\Omega/\varrho} + \sqrt{mq(\epsilon + \sqrt{\epsilon_\Omega/\rho})/\varrho}\right)$.

We see that the proposed FPME procedure is robust to noise, and its query complexity depends linearly in the number of unknown entities. For instance, line 2 in Algorithm 5.1 elicits $\hat{\mathbf{a}} \in \mathbf{R}^q$ by posing $\tilde{O}(q)$ queries, the 'for' loop in line 9 of Algorithm 5.1 runs for $\binom{m}{2}$ iterations, where each iteration requires $\tilde{O}(2q)$ queries, and finally line 16 in Algorithm 5.1 is a simple binary search requiring $\tilde{O}(1)$ queries. The work in Chapter 4 work suggests that linear multiclass elicitation (LPME) elicits misclassification costs (ϕ) with linear query complexity. Surprisingly, our proposed FPME procedure elicits a more complex (nonlinear) metric without increasing the query complexity order. Furthermore, since sample estimates of rates are consistent estimators, and the metrics discussed are 1-Lipschitz wrt. rates, with high probability, we gather correct oracle feedback from querying with finite sample estimates $\Omega(\hat{\mathbf{r}}_1^{1:m}, \hat{\mathbf{r}}_2^{1:m})$ instead of querying with population statistics $\Omega(\mathbf{r}_1^{1:m}, \mathbf{r}_2^{1:m})$, as long as we have sufficient samples. Apart from this, Algorithm 1 is agnostic to finite sample errors as long as the sphere $\mathcal{S}_\rho$ is contained within the feasible region $\mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$.



(a) (b) (c)

Figure 5.3: Elicitation error in recovering the oracle's metric.

5.5 EXPERIMENTS

5.5.1 Theory Validation

We first empirically validate the FPME procedure and recovery guarantees of Section 5.4. Recall that there exists a sphere $\mathcal{S}_\rho \subset \mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$ as long as there is a non-trivial classification signal within each group (Remark 5.2). Thus for experiments, we assume access to a feasible sphere $\mathcal{S}_\rho$ with $\rho = 0.2$. We randomly generate 100 oracle metrics each for $k, m \in \{2, 3, 4, 5\}$ parametrized by $\{\bar{\mathbf{a}}, \overline{\mathbf{B}}, \bar{\lambda}\}$. This specifies the query outputs by the oracle for each metric in Algorithm 5.1. We then use Algorithm 5.1 with tolerance $\epsilon = 10^{-3}$ to elicit corresponding metrics parametrized by $\{\hat{\mathbf{a}}, \hat{\mathbf{B}}, \hat{\lambda}\}$. Algorithm 5.1 makes $1 + 2M$ subroutine calls to LPME procedure and 1 call to Algorithm 5.2. LPME subroutine requires exactly $16(q-1)\log(\pi/2\epsilon)$ queries, where we use 4 queries to shrink the interval in the binary search loop and fix 4 cycles for the coordinate-wise search. Also, Algorithm 5.2 requires $4\log(1/\epsilon)$ queries. In Figure 5.3, we report the mean of the ℓ_2-norm between the oracle's metric and the elicited metric. Clearly, we elicit metrics that are close to the true metrics. Moreover, this holds true across a range of m and k values demonstrating the robustness of the proposed approach. Figure 5.3(a) shows that the error $\|\bar{\mathbf{a}} - \hat{\mathbf{a}}\|_2$ increases only with the number of classes k and not groups m. This is expected since $\hat{\mathbf{a}}$ is elicited by querying rates that zero out the fairness violation (Section 5.3.1). Figure 5.3(b) verifies Theorem 5.1 by showing that $\|\text{vec}(\overline{\mathbf{B}}) - \text{vec}(\hat{\mathbf{B}})\|_2$ increases with both number of classes k and groups m. In accord with Theorem 5.1, Figure 5.3(c) shows that the elicited trade-off $\hat{\lambda}$ is also close to the true $\bar{\lambda}$. However, the elicitation error increases consistently with groups m but not with classes k. A possible reason may be the cancellation of errors from eliciting $\hat{\mathbf{a}}$ and $\hat{\mathbf{B}}$ separately.

Table 5.1: Dataset statistics; the real-valued regressor in *wine* and *crime* datasets is recast to 3 classes based on quantiles.

Dataset	k	m	**#samples**	**#features**	**group.feat**
default	2	2	30000	33	gender
adult	2	3	43156	74	race
wine	3	2	6497	13	color
crime	3	3	1907	99	race

Table 5.2: Common (baseline) metrics usually deployed to rank classifiers.

Name →	$\hat{\phi}\hat{\varphi}\hat{\lambda}$_a	$\hat{\phi}\hat{\varphi}\hat{\lambda}$_w	$\hat{\phi}\hat{\varphi}$_a	$\hat{\phi}\hat{\varphi}$_w	$\hat{\phi}$_a	$\hat{\phi}$_w	o_p	o_f
$\hat{\mathbf{a}}$	acc.	w-acc.	acc.	w-acc.	acc.	w-acc.	$\bar{\mathbf{a}}$	-
$\hat{\mathbf{B}}$	acc.	w-acc.	acc.	w-acc.	elicit	elicit	-	$\overline{\mathbf{B}}$
$\hat{\lambda}$	0.5	w-acc.	elicit	elicit	elicit	elicit	0	1

5.5.2 Ranking of Classifiers

Next, we highlight the utility of FPME in ranking real-world classifiers. One of the most important applications of performance metrics is evaluating classifiers, i.e., providing a quantitative score for their quality which then allows us to choose the best (or best set of) classifier(s). In this section, we discuss how the ranking of plausible classifiers is affected when a practitioner employs default metrics to rank (fair) classifiers instead of the oracle's metric or our elicited approximation.

We take four real-world classification datasets with $k, m \in \{2, 3\}$ (see Table 5.1). 60% of each dataset is used for training and the rest for testing. We create a pool of 100 classifiers for each dataset by tweaking hyperparameters under logistic regression models [57], multi-layer perceptron models [58], support vector machines [59], LightGBM models [60], and fairness constrained optimization based models [61]. We compute the group wise confusion rates on the test data for each model for each dataset. We will compare the ranking of these classifiers achieved by competing baseline metrics with respect to the ground truth ranking.

We generate 100 random oracle metrics $\overline{\Psi}$. $\overline{\Psi}$'s gives us the ground truth ranking of the above classifiers. We then use our proposed procedure FPME (Algorithm 5.1) to recover the oracle's metric. For comparison in ranking of real-world classifiers, we choose a few metrics that are routinely employed by practitioners as baselines (see Table 5.2). The prefixes (i.e., $\hat{\phi}, \hat{\varphi}$, or $\hat{\lambda}$) in name of the baseline metrics denote the components that are set to default metrics, and the suffixes (i.e. 'a' or 'wa') denote whether the assignment is done with *accuracy* (i.e., equal weights) or with *weighted accuracy* (weights are assigned randomly however maintaining the true order of weights as in $\overline{\Psi}$). For example, $\hat{\phi}\hat{\varphi}\hat{\lambda}$_a corresponds to

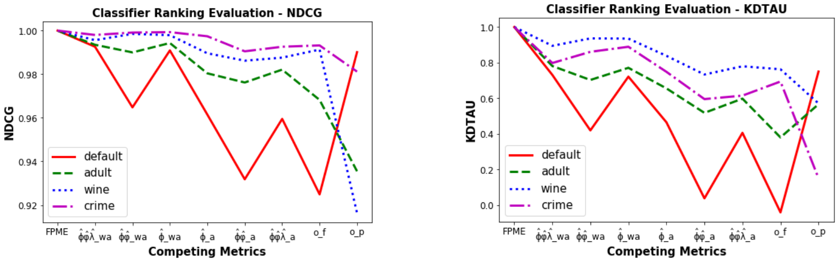


Figure 5.4: Ranking performance of real-world classifiers by competing metrics.

the metric where $\hat{\phi}, \hat{\varphi}, \hat{\lambda}$ are set to standard classification accuracy. Similarly, $\hat{\phi}$_w denote a metric where the misclassification cost $\hat{\phi}$ is set to weighted accuracy but both $\hat{\varphi}$ and $\hat{\lambda}$ are elicited using Part 2 and Part 3 of the FPME procedure (Algorithm 5.1), respectively. Assigning weighted accuracy versions is a commonplace since sometimes the order of the costs associated with the types of mistakes in misclassification cost $\overline{\phi}$ or fairness violation $\overline{\varphi}$ or preference for fairness violation over misclassification $\overline{\lambda}$ is known but not the actual cost. Another example is $\hat{\phi}\hat{\varphi}$_a which corresponds to the metric where $\hat{\phi}, \hat{\varphi}$ are set to accuracy and only the trade-off $\hat{\lambda}$ is elicited using Part 3 of the FPME procedure (Algorithm 5.1). This is similar to prior work by Zhang et al. [22] who assumed the classification error and fairness violation known, so only the trade-off has to be elicited – however they also assume direct ratio queries, which can be challenging in practice. Our approach applies much simnpler pairwise preference queries. Lastly, o_p and o_f represent *only predictive performance* with $\lambda = 0$ and *only fairness* with $\lambda = 1$, respectively.

Figure 5.4 shows average NDCG (with exponential gain) [62] and Kendall-tau coefficient [63] over 100 metrics $\overline{\Psi}$ and their respective estimates by the competing baseline metrics. We see that FPME, wherein we elicit $\hat{\phi}, \hat{\varphi}$, and $\hat{\lambda}$ in sequence, achieves the highest possible NDCG and Kendall-tau coefficient. Even though we make some elicitation error in recovery (Section 5.4), we achieve almost perfect results while ranking the classifiers.

To connect to practice, this implies that when given a set of classifiers, ranking based on elicited metrics will align most closely to ranking based on the true metric, as compared to ranking classifiers based on default metrics. This is a crucial advantage of metric elicitation for practical purposes. In this experiment, baseline metrics achieve inferior ranking of classifiers in comparison to the rankings achieved by metrics that are elicited using the proposed FPME procedure. Figure 5.4 also suggests that it is beneficial to elicit all three components $(\overline{\mathbf{a}}, \overline{\mathbf{B}}, \overline{\lambda})$ of the metric in Definition 5.1, rather than pre-define a component and elicit the rest. For the *crime* dataset, some methods also achieve high NDCG values, so ranking at the top is good; however Kendall-tau coefficient is weak which suggests that overall ranking is poor. With the exception of the *default* dataset, the weighted versions are better than equally weighted versions in ranking. This is expected because in weighted versions, at least order of the preference for the type of costs matches with the oracle's preferences.

5.6 RELATED WORK

Some early attempts to eliciting individual fairness metrics [64, 65] are distinct from ours – as we are focused on the more prevalent setting of group fairness, yet for which there are no existing approaches to our knowledge. Zhang et al. [22] propose an approach that elicits only the trade-off between accuracy and fairness using complicated ratio queries. We, on the other hand, elicit classification cost, fairness violation, and the trade-off together as a non-linear function, all using much simpler pairwise comparison queries. Prior work for constrained classification focus on learning classifiers under constraints for fairness [15, 16, 66, 67]. We take the regularization view of algorithmic fairness, where a fairness violation is embedded in the metric definition instead of as constraints [11, 50, 52, 55, 68]. From the elicitation perspective, the closest line of work to ours is in Chapters 3 and 4, where we proposed the problem of ME but solved it only for a simpler setting of classification without fairness. As we move to multiclass, multigroup fair performance ME, we find that the complexity of both the form of the metrics and the query space increases. This results in starkly different elicitation strategy with novel methods required to provide query complexity guarantees. Learning (linear) functions passively using pairwise comparisons is a mature field [31, 34, 40], but these approaches fail to control sample (i.e. query) complexity. Active learning in fairness [69] is a related direction; however the aim there is to learn a fair classifier based on fixed metric instead of eliciting the metric itself.

5.7 CONCLUDING REMARKS AND FUTURE WORK

- **Transportability:** Our elicitation procedure is independent of the population $\mathbb{P}$ as long as there exists a sphere of rates which is feasible for all groups. Thus, any metric that is learned using one dataset or model class (i.e., by estimated $\hat{\mathbb{P}}$) can be applied to other applications and datasets, as long as the expert believes the context and tradeoffs are the same.

- **Extensions.** Our propsal can be modified to leverage the structure in the metric or the groups to further reduce the query complexity. For example, when the fairness violation weights are the same for all pairs of groups, the procedure in Section 5.3.2

requires only one partitioning of groups to elicit the metric $\hat{\varphi}$. Such modifications are easy to incorporate. In the future, we plan to extend our approach to more complex metrics such as linear-fractional functions of rates and discrepancies.

- **Limitations of group-fair metrics.** Since the metrics we consider depend on a classifier only through its rates, comparing two classifiers on these metrics is equivalent to comparing their rates. Unfortunately, with this setup, all the limitations associated with group-fairness definition of metrics apply to our setup as well. For example, we may discard notions of *individual fairness* when only group-rates are considered for comparing classifiers [70]. Similarly, issues associated with *overlapping groups* [71], *detailed group specification* [71], *unknown or changing groups* [72, 73], *noisy or biased* group information [74], among others, pose limitations to our proposed setup. We hope that as the first work on the topic, our work will inspire the research community to address many of these open problems for the task of metric elicitation.

- **Optimal bounds.** We conjecture that our query complexity bounds are tight; however, we leave this detail for the future. In conclusion, we elicit a more complex (non-linear) group fair-metric with the same query complexity order as standard classification linear elicitation procedures (Chapter 4).

- **Limitation.** Our work seeks to truly democratize and personalize fair machine learning. Besides, the significance of fair performance metric elicitation lies in how it empowers the practitioner to tune the design of machine learning models to the needs of the target fairness task. However, at the same time, this work may have drawbacks because it leaves open the key question of who should be the stakeholders to be queried. This work also assumes a parametric form for the oracle metric, which may not be an exact match to practice. Furthermore, we should be cautious of the result of the failure of the system which could cause disparate impact among sensitive groups when the elicited metric is incorrect, e.g., when applied to settings where the stated assumptions are not met.

CHAPTER 6: QUADRATIC METRIC ELICITATION FOR FAIRNESS AND BEYOND

The Metric Elicitation (ME) strategies for the binary and multiclass classification setups that are discussed in Chapters 3 and 4, respectively, only handle linear or quasi-linear function of predictive rates, which can be restrictive for many applications where the metrics are complex and non-linear. For example, in *fair machine learning*, classifiers are often judged by measuring discrepancies between predictive rates for different protected groups [16]. Similar discrepancy-based measures are also used in *distribution matching* applications [15, 75]. A common measure of discrepancy in such applications is the squared difference, which is appealing for its smoothness properties and a quadratic metric that cannot be handled by existing approaches. Similar quadratic metrics also find use in class-imbalanced learning [15, 66] (see Section 6.1.3 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 [76]. This allows one to better capture real-world human preferences.

Our high-level idea is to approximate the quadratic metric using 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 pairwise queries, we only receive *slopes (directions)* and not magnitudes for the local-linear functions, requiring intricate analysis 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 unknown entities, which we show is *near-optimal.*

Our interest in quadratic metric elicitation is majorly motivated by applications to *fair machine learning* [9, 13, 16]. While several group-based fairness metrics have been proposed to capture bias in automated decision-making, selecting the right metric remains a crucial challenge [22]. In Chapter 5, we proposed an approach for eliciting group-fair metrics that measure discrepancies using the absolute differences in rates across multiple sensitive groups. Unfortunately, that approach specifically handles metrics that are linear in the group discrepancies and does not generalize easily 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 metrics. Like we did in Chapter 5, here we jointly elicit three terms: (i) predictive performance defined by a weighted error metric, (ii) a quadratic fairness violation metric, and (iii) a trade-off between the predictive performance and fairness violation.

Contributions and chapter organization. We propose a novel quadratic metric elicitation algorithm for classification problems, which requires only pairwise preference feedback either over classifiers or rates (Section 6.2). Specific to group-based fairness tasks, we show how to jointly elicit the predictive and fairness metrics, and the trade-off between them (Section 6.3). The proposed approach is robust under feedback and finite sample noise and requires a near-optimal number of queries for elicitation (Section 6.4). We empirically validate the proposal for multiple classes and groups on simulated oracles (Section 6.5). Lastly, we discuss how our strategy can be generalized to elicit higher-order polynomials by recursively applying the procedure to elicit lower-order approximations (Section 6.6). All the proofs in this chapter are provided in Appendix D.

Notation. $\|\cdot\|_F$ 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.

6.1 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

$$h : \mathcal{X} \to \Delta_k \tag{6.1}$$

that for any $\mathbf{x}$ gives a distribution $h(\mathbf{x})$ over the k classes and use

$$\mathcal{H} = \{h : \mathcal{X} \to \Delta_k\} \tag{6.2}$$

to denote the set of all classifiers. Unlike Chapter 4, our choice of measurement space is the space of predictive rates (described next). This is just to suit the application of fairness, where predictive rates for two sensitive groups can be compared; however, it is not suitable for group-fair application purposes to compare confusion matrix entries for two sensitive groups. Nevertheless, the proposed algorithm for quadratic (or, polynomial) metric elicitation will also work if the choice of measurement space is the space of confusion matrices.

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:

$$R_{ij}(h, \mathbb{P}) \coloneqq P(h(X) = j | Y = i) \quad \text{for } i, j \in [k], \tag{6.3}$$

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:

$$R_{ii}(h, \mathbb{P}) = 1 - \sum\nolimits_{j=1, j \neq i}^{k} R_{ij}(h, \mathbb{P}). \tag{6.4}$$

Thus, we can represent a rate matrix with its $q \coloneqq (k^2 - k)$ 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 rates'*.

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

$$\phi(\mathbf{r}(h, \mathbb{P})). \tag{6.5}$$

This includes the (weighted) error rate $\phi^{\text{err}}(\mathbf{r}(h, \mathbb{P})) = \sum_i a_i r_i(h, \mathbb{P})$, for weights $a_i \in \mathbb{R}_+$, the F-measure, and many more metrics [5]. Without loss of generality (w.l.o.g.), we treat metrics as costs. Since the metric's scale does not affect the learning problem [18], we allow $\phi : [0, 1]^q \to [-1, 1]$.

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:

$$\mathcal{R} = \{\mathbf{r}(h, \mathbb{P}) \,:\, h \in \mathcal{H}\}. \tag{6.6}$$

For simplicity, we will suppress the dependence on $\mathbb{P}$ and h if it is clear from the context.

6.1.1 Metric Elicitation: Problem Setup

We now describe the problem of *Metric Elicitation*, which follows from Chapter 2. 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 provides answers by comparing its value on the two classifiers.

Definition 6.1 (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:

$$\Gamma(h_1, h_2\,; \phi) = \Omega(\mathbf{r}_1, \mathbf{r}_2\,; \phi) = \mathbf{1}[\phi(\mathbf{r}_1) > \phi(\mathbf{r}_2)], \tag{6.7}$$

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 [19, 20] or via A/B testing [21]. 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., [22, 23]), and have the user provide the preference. 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 D.1.1 for details). We next formally state the ME problem.

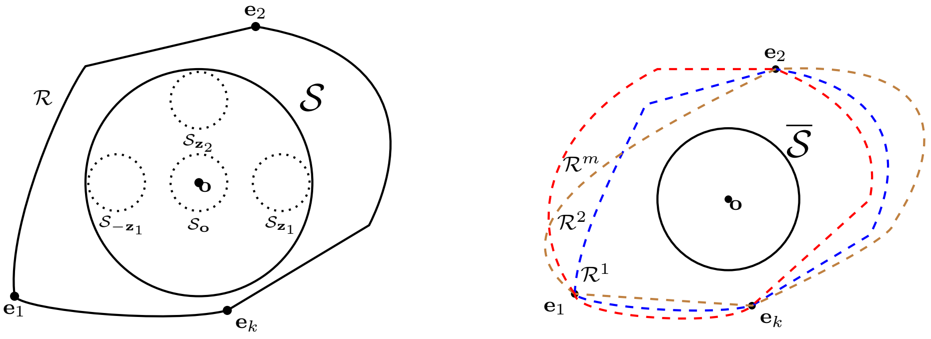
Definition 6.2 (Metric Elicitation with Pairwise Queries (given $\{(\mathbf{x}, y)_i\}_{i=1}^n$))**.** Suppose that the oracle's (unknown) performance metric is ϕ. Using oracle queries of the form $\Omega(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2\,;\, \phi)$, where $\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2$ 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 $\kappa > 0$.

As discussed in previous chapters, the performance of ME is evaluated both by the query complexity and the quality of the elicited metric. As is standard in the decision theory literature [6, 17, 24], 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 $\hat{\mathbf{r}}(h, \{(\mathbf{x}, y)_i\}_{i=1}^n)$.

6.1.2 Linear Metric Elicitation

As a warm up, we overview the Linear Performance Metric Elicitation (LPME) procedure of Chapter 4, which we will use as a subroutine. Here we assume that the oracle's metric is a linear function of rates $\phi^{\text{lin}}(\mathbf{r}) \coloneqq \langle \mathbf{a}, \mathbf{r} \rangle$, for some unknown costs $\mathbf{a} \in \mathbf{R}^q$. In other words, given two rates $\mathbf{r}_1$ and $\mathbf{r}_2$, the oracle returns $\mathbf{1}[\langle \mathbf{a}, \mathbf{r}_1 \rangle > \langle \mathbf{a}, \mathbf{r}_2 \rangle]$. Since the metrics are scale invariant [17, 18], 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 simple one-dimensional binary search. When $k > 2$, one can apply a coordinate-wise procedure,



(a) (b)

Figure 6.1: (a) Geometry of 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) enclosing a common sphere $\overline{\mathcal{S}} \subset \mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$.

performing a binary search in one coordinate, while keeping the others fixed. The efficacy of this procedure, however, hinges on the geometry of the underlying set of feasible rates $\mathcal{R}$, which we discuss below. We first make a mild assumption ensuring that there is some signal for non-trivial classification.

Assumption 6.1. The conditional-class distributions are distinct, i.e., $P(Y = i|X) \neq P(Y = j|X) \;\; \forall\, i, j \in [k]$.

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

Proposition 6.1 (Geometry of $\mathcal{R}$; Figure 6.1(a))**.** The set of rates $\mathcal{R} \subseteq [0, 1]^q$ 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 a classifier which for any input predicts each class with equal probability.

Remark 6.1 (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 equivalently 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 $O(q \log(1/\epsilon))$ queries recovers coefficients $\hat{\mathbf{a}}$ with $\|\mathbf{a} - \hat{\mathbf{a}}\|_2 \leq O(\sqrt{q}\epsilon)$. Please see Chapter 4 for details.

Remark 6.2 (LPME Guarantee)**.** Given any q-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 4.2, Chapter 4) 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., ${}^{a_i}/_{a_j} \approx {}^{\hat{a}_i}/_{\hat{a}_j} \; \forall \; i, j \in [q]$.

Note that the algorithm is closely tied with the scale invariance condition and thus only estimates the direction (slope) of the coefficient vector $\mathbf{a}$, and not its magnitude. 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 D.1.1, we discuss an efficient procedure for identifying a sphere of suitable radius.

6.1.3 Quadratic Performance Metrics

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

Definition 6.3 (Quadratic Metric)**.** For a vector $\mathbf{a} \in \mathbb{R}^q$ and a symmetric matrix $\mathbf{B} \in \mathbb{R}^{q \times q}$ with $\|\mathbf{a}\|_2^2 + \|\mathbf{B}\|_F^2 = 1$ (wlog. due to scale invariance):

$$\phi^{\text{quad}}(\mathbf{r}\,;\,\mathbf{a}, \mathbf{B}) = \langle \mathbf{a}, \mathbf{r} \rangle + \frac{1}{2}\mathbf{r}^T\mathbf{B}\mathbf{r}. \tag{6.8}$$

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

Example 6.1 (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 [14, 77, 78], which is the quadratic mean of rates:*

$$\phi^{\text{qmean}}(\mathbf{r}) \;=\; 1/k \sum_{i=1}^{k} \left(\sum_{j=1}^{k-1} r_{(i-1)(k-1)+j} \right)^2. \tag{6.9}$$

Example 6.2 (Distribution matching)**.** *In certain applications, one needs the proportion of predictions for each class (i.e., the coverage) to match a target distribution $\boldsymbol{\pi} \in \Delta_k$ [15, 61, 66, 79]. A measure often used for this task is the squared difference between the per-class coverage and the target distribution:*

$$\phi^{\text{cov}}(\mathbf{r}) \;=\; \sum_{i=1}^{k} \big(\text{cov}_i(\mathbf{r}) - \pi_i\big)^2, \tag{6.10}$$

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*

$\mathbb{P}(Y = i)$ [75, 80]. We capture more general quadratic distance measures for distributions, e.g.,

$$(\text{cov}(\mathbf{r}) - \boldsymbol{\pi})^{\mathbf{T}}\mathbf{Q}(\text{cov}(\mathbf{r}) - \boldsymbol{\pi}) \tag{6.11}$$

for a positive semi-definite matrix $\mathbf{Q} \in PSD_k$ [81].

Example 6.3 (Fairness violation). *A popular criterion for group-based fairness is equalized odds, which requires equal rates across different protected groups [16, 55]. 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:*

$$\phi^{\text{EO}}((\mathbf{r}^1, \dots, \mathbf{r}^m)) = \sum_{v>u}\sum_{i=1}^{q} (r_i^u - r_i^v)^2 \,. \tag{6.12}$$

Other quadratic fair-criteria for two classes include equal opportunity $\phi^{EOpp}((\mathbf{r}^1, \dots, \mathbf{r}^m)) = \sum_{v>u}(r_1^u - r_1^v)^2$ [16], balance for the negative class $\phi^{BN}((\mathbf{r}^1, \dots, \mathbf{r}^m)) = (r_2^u - r_2^v)^2$ [13], error-rate balance $\phi^{EB}((\mathbf{r}^1, \dots, \mathbf{r}^m)) = 0.5\sum_{v>u}(r_1^u - r_1^v)^2 + (r_2^u - r_2^v)^2$ [48], etc. and their weighted variants. In Section 6.3, we consider metrics that trade-off between an error term and a quadratic fairness term.

Note that, due to the scale invariance condition in Definition 6.3, 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. Lastly, we need the following assumption on the metric.

Assumption 6.2. 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 convex ϕ^{quad}, where it merely implies that the optimal classifier for the metric is not the uniform random classifier.

6.2 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 6.3) 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 rate vectors and invokes LPME to estimate the local-linear approximations' slopes. The challenge, of course, 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.

6.2.1 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 6.1):

$$\begin{aligned}\phi^{\text{quad}}(\mathbf{r};\, \mathbf{a}, \mathbf{B}) &= \langle \mathbf{d}, \mathbf{r} - \mathbf{o} \rangle + \frac{1}{2}(\mathbf{r} - \mathbf{o})^T \mathbf{B} (\mathbf{r} - \mathbf{o}) + c \\ &= \overline{\phi}(\mathbf{r};\, \mathbf{d}, \mathbf{B}) + c, \end{aligned} \tag{6.13}$$

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 $\overline{\phi}(\mathbf{r};\, \mathbf{d}, \mathbf{B})$.

Let z be a fixed point in $\mathcal{R}$. Since the metric in Definition 6.3 is smooth, the metric can be closely approximated by its first-order Taylor expansion in a small neighborhood around $\mathbf{z}$, i.e.,

$$\overline{\phi}(\mathbf{r};\, \mathbf{d}, \mathbf{B}) \approx \langle \mathbf{d} + \mathbf{B}(\mathbf{z} - \mathbf{o}), \mathbf{r} \rangle + c', \tag{6.14}$$

for a constant c'. So if we apply LPME to the metric $\overline{\phi}$ 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 will 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 (6.8). For simplicity, we will assume that the oracle is noise-free and later show robustness to noise and the query complexity guarantees in Section 6.4.

6.2.2 Eliciting Metric Coefficients

We outline the main steps of Algorithm 6.1 below. Please see Appendix D.2 for the full derivation.

Estimate coefficients d (Line 2). We first wish to estimate the linear portion $\mathbf{d}$ of the metric $\overline{\phi}$ in (6.13). 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 $\mathbf{o}$. See Figure 6.1(a) for an illustration. Within this ball, the metric $\overline{\phi}$ approximately equals the linear function $\langle \mathbf{d}, \mathbf{r} \rangle + c'$ using (6.14), and so the LPME gives us an estimate of the slope of $\mathbf{d}$. From Remark 6.2, the estimates $\mathbf{f}_0 = (f_{10}, \ldots, f_{q0})$

Algorithm 6.1 QPM Elicitation

```
1: Input: $\mathcal{S}$, Search tolerance $\epsilon > 0$, Oracle $\Omega$ with metric $\overline{\phi}$
2: $\mathbf{f}_0 \leftarrow$ LPME($\mathcal{S}_{\mathbf{o}}, \epsilon, \Omega$) with $\mathcal{S}_{\mathbf{o}} \subset \mathcal{S}$ and obtain (6.15)
3: for $j \in \{1, 2, \dots, q\}$ do
4:     $\mathbf{f}_j \leftarrow$LPME$\big(\mathcal{S}_{\mathbf{z}_j}, \epsilon, \Omega\big)$ with $\mathcal{S}_{\mathbf{z}_j} \subset \mathcal{S}$ and obtain (6.16)
5: end for
6: $\mathbf{f}_1^- \leftarrow$ LPME($\mathcal{S}_{-\mathbf{z}_1}, \epsilon, \Omega$) with $\mathcal{S}_{-\mathbf{z}_1} \subset \mathcal{S}$ and obtain (6.17)
7: $\hat{\mathbf{a}}, \hat{\mathbf{B}} \leftarrow$ normalized solution dervied from (6.18)
8: Output: $\hat{\mathbf{a}}, \hat{\mathbf{B}}$
```

approximately satisfy the following $(q-1)$ equations:

$$\frac{d_i}{d_1} = \frac{f_{i0}}{f_{10}} \qquad \forall\, i \in \{2, \dots, q\}. \tag{6.15}$$

Estimate coefficients B (Lines 3–5). Next, we wish to estimate each column of the matrix $\mathbf{B}$ of the metric $\overline{\phi}$ in (6.13). For this, we apply LPME to small neighborhoods around points in the direction of standard basis vectors $\boldsymbol{\alpha}_j \in \mathbb{R}^q$, $j = 1, \dots, q$. 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 6.1(a) for the visual intuition. The returned estimates $\mathbf{f}_j = (f_{1j}, \dots, f_{qj})$ approximately satisfy:

$$\frac{d_i + (\rho - \varrho)B_{ij}}{d_1 + (\rho - \varrho)B_{1j}} = \frac{f_{ij}}{f_{1j}} \quad \forall\, i \in \{2, \dots, q\},\ j \leq i. \tag{6.16}$$

Since the matrix $\mathbf{B}$ is symmetric, so far we have $q(q+1)/2$ equations. Now note that since we are only eliciting slopes using LPME, we always lose out on one degree of freedom. Hence, there are q more unknown entities, and to estimate them we need $q - 1$ more equations beside the one 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 6.1(a). The returned slopes $\mathbf{f}_1^- = (f_{11}^-, \dots, f_{q1}^-)$ approximately satisfy:

$$\frac{d_2 - (\rho - \varrho)B_{21}}{d_1 - (\rho - \varrho)B_{11}} = \frac{f_{21}^-}{f_{11}^-}. \tag{6.17}$$

Put together (Line 6). By combining (6.15), (6.16) and (6.17), we express each entry of

$\mathbf{B}$ in terms of d_1:

$$B_{ij} = \Big(F_{i,1,j}(1 + F_{j,1,1}) - F_{i,1,j}F_{j,1,0}d_1 - F_{i,1,0} + F_{i,1,j}\tfrac{F^-_{2,1,1} + F_{2,1,1} - 2F_{2,1,0}}{F^-_{2,1,1} - F_{2,1,1}} \Big) d_1, \tag{6.18}$$

where $F_{i,j,l} = f_{il}/f_{jl}$ and $F^-_{i,j,l} = f^-_{il}/f^-_{jl}$. 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. Moreover, the derivation so far assumes $d_1 \neq 0$. This is based on Assumption 6.2 which states that at least one coordinate of $\mathbf{d}$ is non-zero, and we've assumed w.l.o.g. that this is d_1. In practice, we can identify a non-zero coordinate using q trivial queries of the form $(\varrho\boldsymbol{\alpha}_i + \mathbf{o}, \mathbf{o}), \forall i \in [q]$.

Here, we emphasize on a key difference with Chapters 3 and 4 which is that, there we relied on a boundary point characterization that does not hold for general nonlinear metrics. Instead, we use structural properties of the metric to estimate local-linear approximations. As we discussed in the beginning of this chapter, while this may seem a natural idea, the QPME procedure tackles three key challenges: (a) works with only *slopes* for the local-linear functions, (b) ensures that the center points for approximations are feasible, and (c) handles the multiplicative errors in the slopes (see Section 6.4).

6.3 ELICITING QUADRATIC FAIRNESS METRICS

We now discuss quadratic metric elicitation for *algorithmic fairness*. We consider the setup of Chapter 5, where the goal is to elicit a metric that trades-off between predictive performance and fairness violation [16, 48, 50, 52, 55]. However, unlike Chapter 5, we handle general quadratic fairness violations and show how QPME can be easily employed to elicit group-fair metrics.

6.3.1 Fairness Preliminaries

We consider a k-class problem comprising m groups and use $g \in [m]$ to denote the group membership. The groups are assumed to be disjoint, fixed, and known apriori [16, 47, 68]. We have access to a dataset of size n denoted by $\{(\mathbf{x}, g, y)_i\}_{i=1}^n$, generated *iid* from a distribution $\mathbb{P}(X, G, Y)$. In this case, we will work with a separate (randomized) classifiers $h^g : \mathcal{X} \rightarrow \Delta_k$ for each group g, and use $\mathcal{H}^g = \{h^g : \mathcal{X} \rightarrow \Delta_k\}$ to denote the set of all classifiers for a group g.

Group predictive rates: Similar to (6.3), we denote the group-conditional rate matrix for a classifier h^g by $\mathbf{R}^g(h^g, \mathbb{P}) \in \mathbb{R}^{k \times k}$, where the ij-th entry is additionally conditioned on a group and is given by:

$$R^g_{ij}(h^g, \mathbb{P}) := \mathbb{P}(h^g = j | Y = i, G = g) \;\; \forall\, i, j \in [k]. \tag{6.19}$$

Analogous to the general setup (Section 6.1), we denote the group rates by vectors $\mathbf{r}^g(h^g, \mathbb{P}) = \textit{off-diag}(\mathbf{R}^g(h^g, \mathbb{P}))$, and the set of feasible rates for group g by

$$\mathcal{R}^g = \{\mathbf{r}^g(h^g, \mathbb{P}) \,:\, h^g \in \mathcal{H}^g\}. \tag{6.20}$$

Rates for overall classifier: We construct the overall classifier $h : (\mathcal{X}, [m]) \rightarrow \Delta_k$ 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 predictive performance of the overall classifier and its fairness violation. For the former, we will measure the overall rate matrix for h as denoted in (6.3), which can also be represented as:

$$R_{ij} := \mathbb{P}(h = j | Y = i) = \sum\nolimits_{g=1}^{m} t^g_i R^g_{ij}, \tag{6.21}$$

where $t^g_i := \mathbb{P}(G = g | Y = i)$ is the prevalence of group g within class i. For the latter, we will need the m group-specific rates, represented together as a tuple:

$$\mathbf{r}^{1:m} := (\mathbf{r}^1, \ldots, \mathbf{r}^m) \in \mathcal{R}^1 \times \cdots \times \mathcal{R}^m =: \mathcal{R}^{1:m}. \tag{6.22}$$

Lastly, the overall rates in (6.21) can be written as a flattened vector $\mathbf{r} \in [0, 1]^q$, and can be expressed in terms of the group-specific rates as $\mathbf{r} = \sum_{g=1}^m \boldsymbol{\tau}^g \odot \mathbf{r}^g$, where $\boldsymbol{\tau}^g := \textit{off-diag}([\mathbf{t}^g \; \mathbf{t}^g \; \ldots \; \mathbf{t}^g])$.

6.3.2 Fair (Quadratic) Metric Elicitation

We seek to elicit a metric that trades-off between predictive performance defined by a linear function of the overall rates $\mathbf{r}$ and fairness violation defined by a quadratic function of the group rates $\mathbf{r}^{1:m}$.

Definition 6.4. *(Fair (Quadratic) Performance Metric)* For misclassification costs $\mathbf{a} \in \mathbb{R}^q$, $\mathbf{a} \geq 0$, fairness violation costs $\mathbb{B} = \{\mathbf{B}^{uv} \in PSD_q\}^m_{u,v=1,v>u}$, and a trade-off parameter $\lambda \in [0, 1]$, we define:

$$\phi^{\text{fair}}(\mathbf{r}^{1:m}; \mathbf{a}, \mathbb{B}, \lambda) \;:=\; (1 - \lambda)\langle \mathbf{a}, \mathbf{r}\rangle \;+\; \lambda \frac{1}{2} \left(\sum\nolimits_{v>u} (\mathbf{r}^u - \mathbf{r}^v)^T \mathbf{B}^{uv} (\mathbf{r}^u - \mathbf{r}^v) \right), \tag{6.23}$$

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 6.1–6.2, we present the problem of fair quadratic metric elicitation.

Definition 6.5 (Fair Quadratic Metric Elicitation with Pairwise Comparison Queries (given $\{(\mathbf{x}, g, y)_i\}_{i=1}^n$))**.** Let Ω be an oracle for the (unknown) metric ϕ^{fair}, which for any given $\mathbf{r}_1^{1:m}, \mathbf{r}_2^{1:m}$, outputs $\Omega(\mathbf{r}_1^{1:m}, \mathbf{r}_2^{1:m}) = \mathbf{1}[\phi^{\text{fair}}(\mathbf{r}_1^{1:m}) > \phi^{\text{fair}}(\mathbf{r}_2^{1:m})]$. Using oracle queries of the form $\Omega(\hat{\mathbf{r}}_1^{1:m}, \hat{\mathbf{r}}_2^{1:m})$, where $\hat{\mathbf{r}}_1^{1:m}, \hat{\mathbf{r}}_2^{1:m}$ are the estimated rates from samples, 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 sufficiently small error tolerance $\kappa > 0$.

Similar to Section 6.1.2, we study the space of feasible rates $\mathcal{R}^{1:m}$ under the following mild assumption.

Assumption 6.3. For each group $g \in [m]$, the conditional-class distributions $P(Y = j|X, G = g)$, $j \in [q]$, are distinct, i.e. there is some signal for non-trivial classification for each group.

Proposition 6.2 (Geometry of $\mathcal{R}^{1:m}$; Figure 6.1(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 6.3 (Existence of sphere $\overline{\mathcal{S}}$ in $\mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$)**.** There exists a sphere $\overline{\mathcal{S}} \subset \mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$ of radius ρ centered at $\mathbf{o}$. Thus, a rate $\mathbf{s} \in \overline{\mathcal{S}}$ 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 separate classifier for each group, the above remark implies that any rate $\mathbf{r}^{1:m} = (\mathbf{s}^1, \ldots, \mathbf{s}^m)$ for arbitrary points $\mathbf{s}^1, \ldots, \mathbf{s}^m \in \overline{\mathcal{S}}$ is achievable for some choice of group-specific classifiers $h^1, \ldots, h^m$. This observation will be useful in the elicitation algorithm we describe next.

6.3.3 Eliciting Metric Parameters $(\mathbf{a}, \mathbb{B}, \lambda)$

We present a strategy for eliciting fair metrics (Definition 6.4) by adapting the QPME algorithm. For simplicity, we focus on the $m = 2$ case and extend our approach to multiple groups in Appendix D.3.

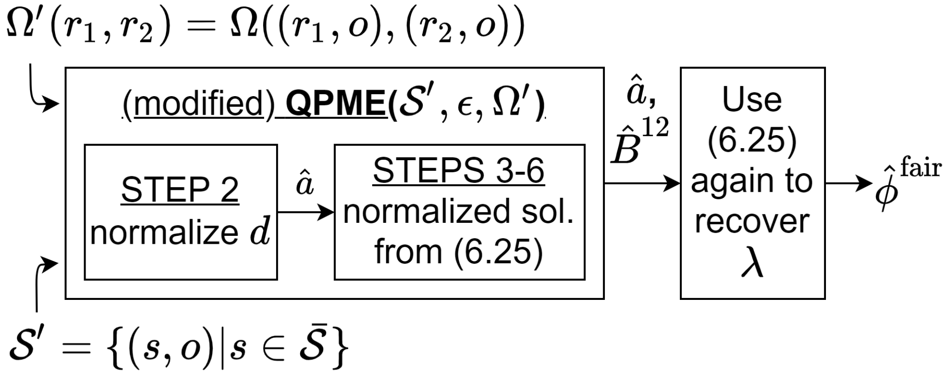

$\Omega'(r_1, r_2) = \Omega((r_1, o), (r_2, o))$
(modified) QPME($\mathcal{S}'$, ϵ, Ω')
STEP 2
normalize d
$\hat{a}$
STEPS 3-6
normalized sol.
from (6.25)
$\hat{a}$,
$\hat{B}^{12}$
Use
(6.25)
again to
recover
λ
$\hat{\phi}^{\text{fair}}$
$\mathcal{S}' = \{(s, o) | s \in \bar{\mathcal{S}}\}$

Figure 6.2: Eliciting Fair Quadratic Metrics (Definition 6.5) for two groups using a minor modification of QPME (Algorithm 6.1).

Observe that for a rate profile $\mathbf{r}^{1:2} = (\mathbf{s}, \mathbf{o})$, where the first group is assigned an arbitrary point in $\overline{\mathcal{S}}$ and the second group is assigned the uniform random classifier's rate $\mathbf{o}$, the fair metric (6.23) becomes:

$$\begin{aligned}\phi^{\text{fair}}((\mathbf{s}, \mathbf{o}); \mathbf{a}, \mathbf{B}^{12}, \lambda) &\coloneqq (1-\lambda)\langle \mathbf{a}, \boldsymbol{\tau}^1 \odot \mathbf{s} + \boldsymbol{\tau}^2 \odot \mathbf{o}\rangle + \frac{\lambda}{2}(\mathbf{s}-\mathbf{o})^T \mathbf{B}^{12}(\mathbf{s}-\mathbf{o}) \\ &\coloneqq \langle \mathbf{d}, \mathbf{s}-\mathbf{o}\rangle + \frac{1}{2}(\mathbf{s}-\mathbf{o})^T \mathbf{B}(\mathbf{s}-\mathbf{o}) \\ &\coloneqq \overline{\phi}(\mathbf{s}; \mathbf{d}, \mathbf{B}),\end{aligned} \tag{6.24}$$

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 $\overline{\phi}$ above is a particular instance of the quadratic metric in (6.13). We can thus apply a slight variant of the QPME procedure in Algorithm 6.1 to solve the quadratic metric elicitation problem over the sphere $\mathcal{S}' = \{(\mathbf{s}, \mathbf{o}) \,|\, \mathbf{s} \in \overline{\mathcal{S}}\}$ 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 7, 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 2 of Algorithm 6.1 and (6.15), 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 3-6 of Algorithm 6.1 and (6.18) gives us:

$$\hat{B}_{ij} = \lambda \hat{B}^{12}_{ij} = \Big(F_{i,1,j}(1 + F_{j,1,1}) - F_{i,1,j}F_{j,1,0}d_1 + F_{i,1,j}\tfrac{F^-_{2,1,1} + F_{2,1,1} - 2F_{2,1,0}}{F^-_{2,1,1} - F_{2,1,1}}\Big)(1-\lambda)\tau_1^1 \hat{a}_1. \tag{6.25}$$

Again by 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 $\hat{B}^{12}_{ij}$ and $\hat{a}_1$, we can now additionally estimate the trade-off parameter $\hat{\lambda}$ from (6.25). See Figure 6.2 for an illustration of the entire procedure.

The proposed approach for the fair (quadratic) metric elicitation easily extends to multiple groups by applying the QPME procedure described above multiple times after fixing one cluster of groups to the rate $\mathbf{o}$ and the remaining to the same rate $\mathbf{s}$ in the intersection sphere $\overline{\mathcal{S}}$. See Appendix D.3 for details. In Appendix D.3.1, we also provide an alternate binary search based method similar to Chapter 5 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. [22], but unlike them, instead of complicated ratio queries, we require simpler pairwise queries.

6.4 GUARANTEES

We discuss guarantees for the QPME procedure (Algorithm 6.1) under the following feedback model, which is useful in practice. The fair metric elicitation guarantees follow directly as a consequence.

Definition 6.6 (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 very close as measured by the metric ϕ^{quad}. Since eliciting the metric involves offline computations including certain ratios, we discuss guarantees under the following regularity assumption that ensures all components are well defined.

Assumption 6.4. For the shifted quadratic metric $\overline{\phi}$ in (6.13), 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 6.1. Given $\epsilon, \epsilon_\Omega \geq 0$, and a 1-Lipschitz metric ϕ^{quad} (Definition 6.3) parametrized by $\mathbf{a}, \mathbf{B}$, under Assumptions 6.1, 6.2, and 6.4, after $O\left(q^2 \log \frac{1}{\epsilon}\right)$ queries Algorithm 6.1 returns a metric $\hat{\phi}^{\text{quad}} = (\hat{\mathbf{a}}, \hat{\mathbf{B}})$ such that $\|\mathbf{a} - \hat{\mathbf{a}}\|_2 \leq O\left(q(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho})\right)$ and $\|\mathbf{B} - \hat{\mathbf{B}}\|_F \leq O\left(q\sqrt{q}(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho})\right)$.

Theorem 6.2. While eliciting the metric ϕ^{quad} (Definition 6.3), at least $\Omega(q^2 \log(1/q\sqrt{q}\epsilon))$ pairwise queries are needed to achieve an error of $q\sqrt{q}\epsilon$ for some (slack) ϵ.

Theorem 6.1 shows that the QPME procedure is robust to noise and its query complexity depends only *linearly* in the number of unknowns. Theorem 6.2 shows that the inherent complexity of the problem is driven by the number of unknowns, which in the most general case (Definition 6.3) is $O(q^2)$. Thus, QPME procedure's query complexity is optimal barring the log term. We stress that despite eliciting a more complex (nonlinear) metric, the query complexity order is same as prior methods for linear elicitation with respect to the number of unknowns [6, 17]. With added structural assumptions on the metric, our proposal can be modified to further reduce the query complexity. For example, suppose one knows that the matrix $\mathbf{B}$ is diagonal, then each LPME subroutine call needs to estimate only one parameter, which can be done in constant number of queries. The resulting query complexity will be $\tilde{O}(q)$ which is again *linear* in the number of unknowns. Moreover, since sample estimates of rates are consistent estimators, and the metrics are 1-Lipschitz w.r.t. rates, with high probability, we gather correct oracle feedback from querying with finite sample estimates $\Omega(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2)$ instead of querying with population statistics $\Omega(\mathbf{r}_1, \mathbf{r}_2)$, as long as we have sufficient samples (see Appendix D.1). Other than this, Algorithm 6.1 is agnostic to finite sample errors as long as the sphere $\mathcal{S}$ is in the space $\mathcal{R}$.

6.5 EXPERIMENTS

We evaluate our approach on simulated oracles. We first present results on a synthetically generated query space and then discuss results on real-world datasets.

6.5.1 Eliciting Metrics

Eliciting quadratic metrics. We first apply QPME (Algorithm 6.1) to elicit quadratic metrics in Definition 6.3. We assume access to a q-dimensional sphere $\mathcal{S}$ centered at rate $\mathbf{o}$ with radius $\rho = 0.2$, from which we query rate vectors $\mathbf{r}$. Recall that in practice, Remark 6.1 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 $k \in \{2, 3, 4, 5\}$ (equiv. $q \in \{2, 6, 12, 20\}$). We run the QPME procedure with tolerance $\epsilon = 10^{-2}$. In Figures 6.3(a)–6.3(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 (and q), showing the effectiveness of our approach in handling multiple classes. The larger standard deviation for $q = 20$ is due to Assumption 6.4 failing to hold in a few trials and the resulting estimates not being as accurate. We discuss this in Section 6.5.2.

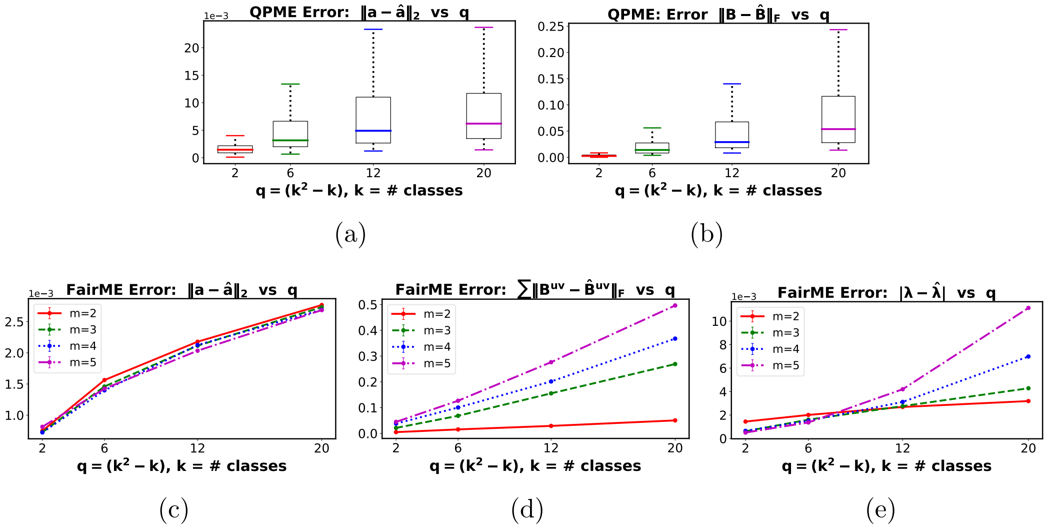


Figure 6.3: Average elicitation error over 100 metrics as a function of number of coefficients q and groups m for quadratic metrics in Definition 6.3 (a–b) and fairness metrics in Definition 6.4 (c–e).

Eliciting fairness metrics. We next apply the elicitation procedure in Figure 6.2 with tolerance $\epsilon = 10^{-2}$ to elicit the fairness metrics in Definition 6.4. 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 6.3(c)–6.3(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 coefficients q 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 q 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.

6.5.2 More Details on Simulated Experiments on Quadratic Metric Elicitation

In Figures 6.3(a)–6.3(b), we show box plots [82] 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.

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 6.4. 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.

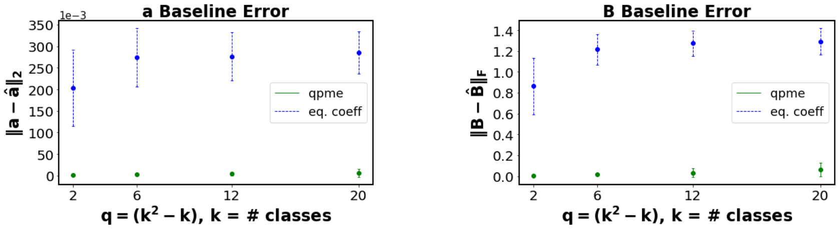


Figure 6.4: Elicitation error in comparison to a baseline which assigns equal coefficients.

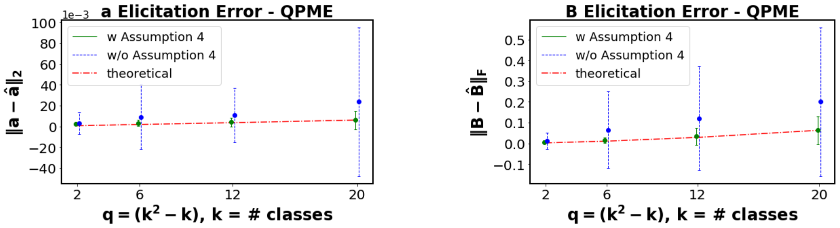


Figure 6.5: Elicitation error for metrics following Assumption 6.4 vs elicitation error for completely random metrics.

Effect of Assumption 6.4. We mentioned in Section 6.5.1 that in a small number of trials, Assumption 6.4 failed to hold with sufficiently large constants $c_0, c_{-1}, c_1 \ldots, c_q$. 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 6.4 effectively ensures that the ratios computed in (6.18) are well-defined. To this end, we generate two sets of 100 quadratic metrics. One set is generated following Assumption 6.4 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.

In Figure 6.5, we see that the elicitation error is much higher when the regularity Assumption 6.4 is not followed, owing to the fact that the ratio computation in (6.18) 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 q (within a constant factor). In Figure 6.5, we see that the mean of ℓ_2 (analogously, Frobenius) norm better follow the theoretical bound trajectory in the case when regularity Assumption 6.4 is followed by the metrics.

We next analyze the ratio of estimated fractions to the true fractions used in (6.18) over 1000 simulated runs. Ideally, this ratio should be 1, but as we see in Figure 6.6, 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 6.4.

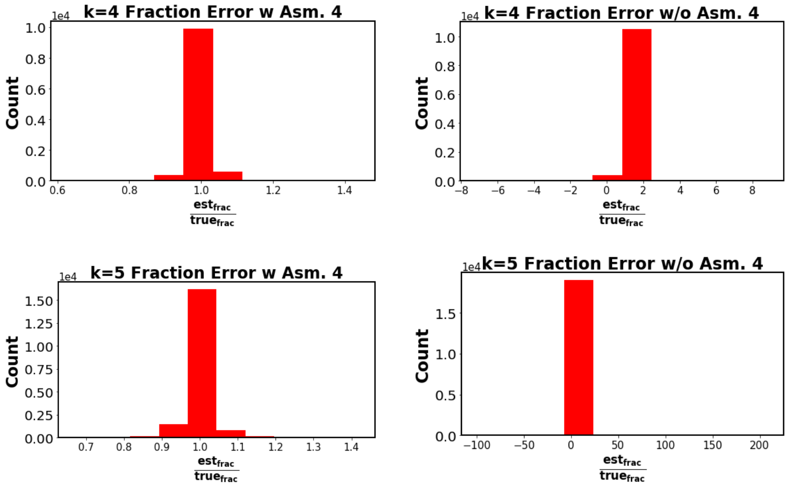


Figure 6.6: Ratio of estimated to true fractions over 1000 simulated runs with and without Assumption 6.4.

Table 6.1: Dataset statistics

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

Since we multiply fractions in (6.18), even then we may observe the compounding effect of fraction estimation errors in the final estimates. Hence, we see for $k = 5$ in Figure 6.3(a)-6.3(b), the standard deviation is high due to few trials where the lower bound of 10^{-2} on the constants in Assumption 6.4 may not be enough. However, majority of the trials as shown in Figure 6.3(a)-6.3(b) and Figure 6.4 incur low elicitation error.

6.5.3 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 metrics while ranking real-world classifiers.

For this experiment, we work with four real world datasets with varying number of classes $k \in \{2, 3, 7\}$. See Table 6.1 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 fa-

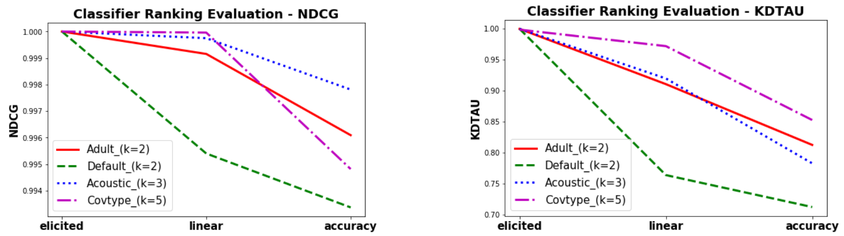


Figure 6.7: Performance of competing metrics while ranking real-world classifiers. 'elicited' is the metric elicited by QPME, 'linear' is the metric that comprises only the linear part of the oracle's true quadratic metric, and 'accuracy' is the linear metric which weigh all classification errors equally (often used in practice).

mous machine learning models that are routinely used in practice. Specifically, we create 20 classifiers each from logistic regression models [57], multi-layer perceptron models [58], LightGBM models [60], and support vector machines [59]. 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 6.3. 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 6.1) 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 $\phi^{acc} = 1/\sqrt{q}\langle \mathbf{1}, \mathbf{r} \rangle$, 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.

We report NDCG (with exponential gain) [62] and Kendall-tau coefficient [63] averaged over the 100 trials in Figure 6.7. 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 6.4, QPME may incur elicitation error, and thus the elicited metrics may not be very accurate; however, Figure 6.7 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.

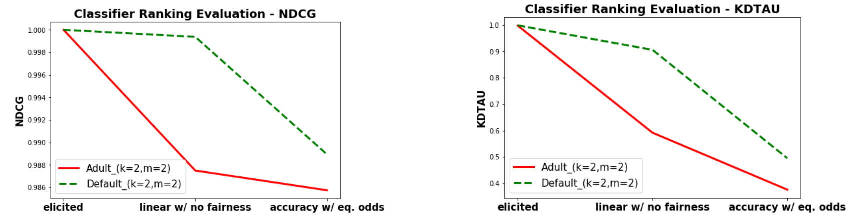


Figure 6.8: Performance of competing metrics while ranking real-world classifiers for fairness. ‘elicited’ is the metric elicited by the (quadratic) fairness metric elicitation procedure from Section 6.3 (also depicted in Figure 6.2), ‘linear w/ no fairness’ is the metric that comprises only the linear part of the oracle’s true quadratic fair metric from Definition 6.4 without the fairness violation, and ‘accuracy w/ eq. odds’ is the metric which weigh all classification errors and fairness violations equally (often used in practice).

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 6.4 that gives ground-truth ranking of classifiers and evaluate the ranking by the elicited (fair-quadratic) metric using the procedure described in Section 6.3 (also depicted in Figure 6.2). In Figure 6.8, we show the NDCG and KD-Tau values for our method and for two baselines: (a) ‘linear w/ no fairness’, which is the metric that comprises only the linear part of the oracle’s true quadratic fair metric from Definition 6.4 without the fairness violation, and (b) ‘accuracy w/ eq. odds’ is the metric which weigh all classification errors and fairness violations equally. We again see that the elicited (fairness) metric’s ranking is closest to the ground-truth.

6.6 EXTENSION TO HIGHER ORDER POLYNOMIALS

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

$$\phi^{\text{cubic}}(\mathbf{r}) \coloneqq \sum_i a_i r_i + \frac{1}{2}\sum_{i,j} B_{ij} r_i r_j + \frac{1}{6}\sum_{i,j,l} C_{ijl} r_i r_j r_l, \tag{6.26}$$

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:

$$\sum_i a_i r_i + \frac{1}{2}\left(\sum_{i,j} B_{ij} r_i r_j + \sum_{i,j,l} C_{ijl}(r_i - z_i)(r_j - z_j) z_l\right) + c, \tag{6.27}$$

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 6.1 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 q times around the basis points $\mathbf{z}_l = \mathbf{o} + (\varrho - \varrho')\boldsymbol{\alpha}_l \;\; \forall l \in [q]$. 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, $\tilde{O}(q^3)$ 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.

6.7 RELATED WORK

Chapter 2 formalized the problem of ME, Chapter 3 put forward an ME procedure for binary classification and then later Chapter 4 extends ME to the multiclass setting [17]. The focus in the previous chapters, however, was on eliciting linear and fractional-linear metrics; whereas, in this chapter, we elicit more complex quadratic metrics. Learning linear functions passively using pairwise comparisons is a mature field [31, 34, 40], but unlike their active learning counter-parts [32, 41, 42], these methods are not query efficient. Other related work include active classification [41, 42, 69], which learn classifiers for a fixed (known) metric. In contrast, we seek to elicit an unknown metric by posing queries to an oracle. There is also some work on active linear elicitation, e.g. Qian et al. [43], but they do not provide theoretical bounds and work with a different query space. We are unaware of prior work on eliciting a quadratic function, either passively or actively using pairwise comparisons.

The use of metric elicitation for fairness is relatively new, with some work on eliciting *individual* fairness metrics [64, 65]. To the best of our knowledge, the work in Chapter 5 is the only work that elicits *group-fair* metrics, which we extend in this chapter to handle more general metrics. Zhang et al. [22] 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, prior work has also focused on learning fair classifiers under constraints [15, 16, 67]. We take the regularization view of fairness, where the fairness violation is included in the objective itself [11, 50, 55, 68].

Our work is also 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 [83, 84] seek to find the optimal decision via minimizing the max-regret over a set of utilities. Studies that directly learn the utility [85, 86] do not provide query complexity guarantees for pairwise comparisons. Formulations that consider a finite set of alternatives [83, 85, 87], 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 [85] or bilinear [86] utilities except for [88] (GAI utilities) and [84] (Choquet integral); whereas, we focus on quadratic metrics which are useful for classification tasks, especially, fairness.

6.8 DISCUSSION, LIMITATIONS, AND FUTURE WORK

We have provided an efficient quadratic metric elicitation strategy and shown its application to the pressing issue in algorithmic fairness. Interestingly, the query complexity for these non-linear metrics has the same dependence on the number of unknowns as that for linear metrics. We have also shown how this idea can be extended to elicit higher order polynomial metrics. This significantly increases the use-cases for ME and opens the door for non-linear metric elicitation. A notable advantage of our proposal is that it is independent of the population $\mathbb{P}$. Thus any metric that is learned using one dataset or model class can be applied to other applications, as long as the expert believes the tradeoffs are the same. A key challenge that we tackle throughout elicitation is maintaining the feasibility of rates, i.e., rates that are achievable by classifiers. This has a practical advantage, because now one has the flexibility to deploy systems that either compare classifiers or compare rates.

At the same time, our work has limitations, too. We assume a parametric form for the quadratic oracle metric, which may not be a good match to practice. Extension to polynomial elicitation helps but may lead to overburdening the oracle with the huge number of queries if the degree of the polynomial is high. Another limitation is that it leaves open the question of who the oracles should be. Furthermore, one should be cautious of the failure of the metric elicitation system especially while eliciting fairness metrics, because that can cause varying impacts among protected groups. We look forward to future work answering these practical questions.

CHAPTER 7: OPTIMIZING BLACK-BOX METRICS THROUGH METRIC ELICITATION

In this chapter, we discuss an interesting application of Metric Elicitation (ME), where the tools and procedures provided in the previous chapters play a key role. We aim to optimize a black-box performance metric, where instead of a *human* oracle, we have a *machine* oracle that responds with absolute quality value of a classifier. As we discuss later, such settings are prevalent in literature. The motivation for using ME for black-box optimization comes from the fact that many existing optimization algorithms are iterative in nature, where in each iteration, they tend to optimize a local-linear approximation. This local-linear approximation of an unknown (black-box) metric can be elicited using the existing ME tools and results. We discuss briefly how these procedures can be extended in the presence of *human* oracles that provide pairwise preference feedback (including the A/B tests based scenarios). We next discuss the formal black-box optimization problem setup and how our tools from ME can be used to optimize metrics in this setup.

7.1 INTRODUCTION

In many real-world machine learning tasks, the evaluation metric one seeks to optimize is not explicitly available in closed-form. This is true for metrics that are evaluated through live experiments or by querying human users [6, 21], or that require access to private or legally protected data [89], and hence cannot be written as an explicit training objective. This is also the case when the learner only has access to data with skewed training distribution or labels with heteroscedastic noise [90, 91], and hence cannot directly optimize the metric on the training set despite knowing its mathematical form.

These problems can be framed as black-box learning tasks, where the goal is to optimize an unknown classification metric on a large (possibly noisy) training data, given access to evaluations of the metric on a small, clean validation sample [91]. Our high-level approach to these learning tasks is to adaptively assign weights to the training examples, so that the resulting weighted training objective closely approximates the black-box metric on the validation sample. We then construct a classifier by using the example weights to post-shift a class-probability estimator pre-trained on the training set. This results in an efficient, iterative approach that does not require any re-training.

Indeed, example weighting strategies have been widely used to both optimize metrics and to correct for distribution shift, but prior works either handle specialized forms of metric or data noise [92, 93, 94], formulate the example-weight learning task as a difficult non-convex problem that is hard to analyze [95, 96], or employ an expensive surrogate re-weighting strategy that comes with limited statistical guarantees [91]. In contrast, we propose a simple and effective approach to optimize a general black-box metric (that is a function of the confusion matrix) and provide a rigorous statistical analysis.

A key element of our approach is eliciting the weight coefficients by probing the black-box metric at few select classifiers and solving a system of linear equations matching the weighted training errors to the validation metric. We choose the "probing" classifiers so that the linear system is well-conditioned, for which we provide both theoretically-grounded options and practically efficient variants. This weight elicitation procedure is then used as a subroutine to iteratively construct the final plug-in classifier.

The contributions in this chapter are as follows:

- We provide a method for eliciting example weights for linear black-box metrics (Section 7.3).
- We use this procedure to iteratively learn a plug-in classifier for general black-box metrics (Section 7.4).
- We provide theoretical guarantees for metrics that are concave functions of the confusion matrix under distributional assumptions (Section 7.5).
- We experimentally show that our approach is competitive with (or better than) the state-of-the-art methods for tackling label noise in CIFAR-10 [97] and domain shift in Adience [98], and optimizing with proxy labels and a black-box fairness metric on Adult [99] (Section 7.7).

All the proofs in this chapter are provided in Appendix E.

Notations: $\text{onehot}(j) \in \{0,1\}^k$ returns the one-hot encoding of $j \in [k]$. In this chapter, the ℓ_2 norm of a vector is denoted by $\|\cdot\|$.

7.2 PROBLEM SETUP

We consider a standard multiclass setup with an instance space $\mathcal{X} \subseteq \mathbb{R}^d$ and a label space $\mathcal{Y} = [k]$. We wish to learn a randomized multiclass classifier $h : \mathcal{X} \to \Delta_k$ that for any input $x \in \mathcal{X}$ predicts a distribution $h(x) \in \Delta_k$ over the k classes. We will also consider deterministic classifiers $h : \mathcal{X} \to [k]$ which map an instance x to one of k classes.

Evaluation Metrics. Let D denote the underlying data distribution over $\mathcal{X} \times \mathcal{Y}$. We will evaluate the performance of a classifier h on D using an evaluation metric $\mathcal{E}^D[h]$, with higher values indicating better performance. Our goal is to learn a classifier h that maximizes this evaluation measure:

$$\max_h \mathcal{E}^D[h]. \tag{7.1}$$

We will focus on metrics $\mathcal{E}^D$ that can be written in terms of classifier's confusion matrix $\mathbf{C}[h] \in [0,1]^{k\times k}$, where the i, j-th entry is the probability that the true label is i and the randomized classifier h predicts j:

$$C^D_{ij}[h] = \mathbf{E}_{(x,y)\sim D}\left[\mathbf{1}(y=i)h_j(x)\right]. \tag{7.2}$$

The performance of the classifier can then be evaluated using a (possibly unknown) function $\psi : [0,1]^{k\times k} \to \mathbb{R}_+$ of the confusion matrix:

$$\mathcal{E}^D[h] = \psi(\mathbf{C}^D[h]). \tag{7.3}$$

Several common classification metrics take this form, including typical linear metrics $\psi(\mathbf{C}) = \sum_{ij} L_{ij}\, C_{ij}$ for some reward matrix $\mathbf{L} \in \mathbb{R}^{k\times k}_+$, the F-measure $\psi(\mathbf{C}) = \sum_i \frac{2C_{ii}}{\sum_j C_{ij} + \sum_j C_{ji}}$ [100], and the G-mean $\psi(\mathbf{C}) = \left(\prod_i \left(C_{ii}/\sum_j C_{ij}\right)\right)^{1/k}$ [101].

We consider settings where the learner has query-access to the evaluation metric $\mathcal{E}^D$, i.e., can evaluate the metric for any given classifier h but cannot directly write out the metric as an explicit mathematical objective. This happens when the metric is truly a black-box function, i.e., ψ is unknown, or when ψ is known, but we have access to only a noisy version of the distribution D needed to compute the metric.

Noisy Training Distribution. For learning a classifier, we assume access to a large sample S^{tr} of n^{tr} examples drawn from a distribution μ, which we will refer to as the "training" distribution. The training distribution μ may be the same as the true distribution D, or may differ from the true distribution D in the feature distribution $\mathbf{P}(x)$, the conditional label distribution $\mathbf{P}(y|x)$, or both. We also assume access to a smaller sample S^{val} of n^{val} examples drawn from the true distribution D. We will refer to the sample S^{tr} as the "training" sample, and the smaller sample S^{val} as the "validation" sample. We seek to solve (7.1) using both these samples.

The following are some examples of noisy training distributions in the literature:

Example 7.1 (Independent label noise (ILN) [93, 94])**.** *The distribution μ draws an example (x, y) from D, and randomly flips y to $\widetilde{y}$ with probability $\mathbf{P}(\widetilde{y}|y)$, independent of the instance x.*

Example 7.2 (Cluster-dependent label noise (CDLN) [102])**.** *Suppose each x belongs to*

Table 7.1: Example weights $\mathbf{W} : \mathcal{X} \to \mathbb{R}_+^{k \times k}$ for linear metric $\mathcal{E}^D[h] = \langle \mathbf{L}, \mathbf{C}^D[h] \rangle$ under the noise models in Exmp. 7.1–7.4, where $W_{ij}(x)$ is the weight on entry C_{ij}. In Sec. 7.3–7.4, we consider metrics that are functions of the diagonal confusion entries alone (i.e. $\mathbf{L}$ and $\mathbf{T}$ are diagonal), and handle general metrics in Appendix E.1.

Model	Noise Transition Matrix	Correction Weights
ILN	$T_{ij} = \mathbf{P}(\widetilde{y} = j \vert y = i)$	$\mathbf{W}(x) = \mathbf{L} \odot \mathbf{T}^{-1}$
CDLN	$T_{ij}^{[m]} = \mathbf{P}(\widetilde{y} = j \vert y = i, g(x) = m)$	$\mathbf{W}(x) = \mathbf{L} \odot (\mathbf{T}^{[g(x)]})^{-1}$
IDLN	$T_{ij}(x) = \mathbf{P}(\widetilde{y} = j \vert y = i, x)$	$\mathbf{W}(x) = \mathbf{L} \odot (\mathbf{T}(x))^{-1}$
DS	-	$W_{ij}(x) = \mathbf{P}^D(x) / \mathbf{P}^\mu(x), \forall i, j$

one of m disjoint clusters $g(x) \in [m]$. The distribution μ draws (x, y) from D and randomly flips y to $\widetilde{y}$ with probability $\mathbf{P}(\widetilde{y} \vert y, g(x))$.

Example 7.3 (Instance-dependent label noise (IDLN) [103])**.** *μ draws (x, y) from D and randomly flips y to $\widetilde{y}$ with probability $\mathbf{P}(\widetilde{y} \vert y, x)$, which may depend on x.*

Example 7.4 (Domain shift (DS) [92])**.** *μ draws $\widetilde{x}$ according to a distribution $\mathbf{P}^\mu(x)$ different from $\mathbf{P}^D(x)$, but draws y from the true conditional $\mathbf{P}^D(y \vert \widetilde{x})$.*

Our approach is to learn example weights on the training sample S^{tr}, so that the resulting weighted empirical objective (locally, if not globally) approximates an estimate of the metric $\mathcal{E}^D$ on the validation sample S^{val}. For ease of presentation, we will assume that the metrics only depend on the diagonal entries of the confusion matrix, i.e., C_{ii}'s. In Appendix E.1, we elaborate how our ideas can be extended to handle metrics that depend on the entire confusion matrix.

While our approach uses randomized classifiers, in practice one can replace them with similarly performing deterministic classifiers using, e.g., the techniques of [104]. In what follows, we will need the empirical confusion matrix on the validation set $\widehat{\mathbf{C}}^{\text{val}}[h]$, where

$$\widehat{C}_{ij}^{\text{val}}[h] = \frac{1}{n^{\text{val}}} \sum_{(x,y) \in S^{\text{val}}} \mathbf{1}(y = i) h_j(x). \tag{7.4}$$

7.3 EXAMPLE WEIGHTING FOR LINEAR METRICS

We first describe our example weighting strategy for linear functions of the diagonal entries of the confusion matrix, which is given by:

$$\mathcal{E}^D[h] \;=\; \textstyle\sum_i \beta_i \, C_{ii}^D[h] \tag{7.5}$$

for some (unknown) weights $\beta_1, \ldots, \beta_k$. In the next section, we will discuss how to use this procedure as a subroutine to handle more complex metrics.

7.3.1 Modeling Example Weights

We define an example weighting function $\mathbf{W} : \mathcal{X} \to \mathbb{R}_+^k$ which associates k *correction weights* $[W_i(x)]_{i=1}^k$ with each example x so that:

$$\mathbf{E}_{(x,y)\sim\mu}\Big[\textstyle\sum_i W_i(x)\,\mathbf{1}(y=i)h_i(x)\Big] \;\approx\; \mathcal{E}^D[h],\; \forall\, h. \tag{7.6}$$

Indeed for the noise models in Examples 7.1–7.4, there exist weighting functions $\mathbf{W}$ for which the above holds with equality. Table 7.1 shows the form of the weighting function for general linear metrics.

Ideally, the weighting function $\mathbf{W}$ assigns k independent weights for each example $x \in \mathcal{X}$. However, in practice, we estimate $\mathcal{E}^D$ using a small validation sample $S^{\text{val}} \sim D$. So to avoid having the example weights over-fit to the validation sample, we restrict the flexibility of $\mathbf{W}$ and set it to a weighted sum of L basis functions $\phi^\ell : \mathcal{X} \to [0,1]$:

$$W_i(x) \;=\; \textstyle\sum_{\ell=1}^L \alpha_i^\ell \phi^\ell(x), \tag{7.7}$$

where $\alpha_i^\ell \in \mathbb{R}$ is the coefficient associated with basis function ϕ^ℓ and diagonal confusion entry (i,i).

In practice, the basis functions can be as simple as a partitioning of the instance space into L clusters, i.e.,:

$$\phi^\ell(x) = \mathbf{1}(g(x) = \ell), \tag{7.8}$$

for a clustering function $g : \mathcal{X} \to [L]$, or may define a more complicated soft clustering using, e.g., radial basis functions [92] with centers x^ℓ and width σ:

$$\phi^\ell(x) = \exp\left(-\|x - x^\ell\|/2\sigma^2\right). \tag{7.9}$$

7.3.2 ϕ-transformed Confusions

Expanding the weighting function in (7.6) gives us:

$$\sum_{\ell=1}^L \sum_{i=1}^k \alpha_i^\ell \; \underbrace{\mathbf{E}_{(x,y)\sim\mu}\big[\phi^\ell(x)\,\mathbf{1}(y=i)h_i(x)\big]}_{\Phi_i^{\mu,\ell}[h]} \approx \mathcal{E}^D[h],\; \forall\, h, \tag{7.10}$$

where $\mathbf{\Phi}^{\mu,\ell}[h] \in [0,1]^k$ can be seen as a ϕ-transformed confusion matrix for the training distribution μ. For example, if one had only one basis function $\phi^1(x) = 1, \forall x$, then $\Phi_i^{\mu,1}[h] = \mathbf{E}_{(x,y)\sim\mu}\big[\mathbf{1}(y = i)h_i(x)\big]$ gives the standard confusion entries for the training distribution. If the basis functions divides the data into L clusters, as in (7.8), then $\Phi_i^{\mu,\ell}[h] = \mathbf{E}_{(x,y)\sim\mu}\big[\mathbf{1}(g(x) = \ell, y = i)h_i(x)\big]$ gives the training confusion entries evaluated on examples from cluster ℓ. We can thus re-write equation (7.6) as a weighted combination of the Φ-confusion entries:

$$\sum_{\ell=1}^{L}\sum_{i=1}^{k} \alpha_i^\ell \Phi_i^{\mu,\ell}[h] \;\approx\; \mathcal{E}^D[h], \forall h. \tag{7.11}$$

7.3.3 Eliciting Weight Coefficients $\boldsymbol{\alpha}$ – The Metric Elicitation Step

We next discuss how to estimate the weighting function coefficients α_i^ℓ's from the training sample S^{tr} and validation sample S^{val}. Notice that (7.11) gives a relationship between statistics $\mathbf{\Phi}^{\mu,\ell}$'s computed on the training distribution μ, and the evaluation metric of interest computed on the true distribution D. Moreover, for a fixed classifier h, the left-hand side is *linear* in the unknown coefficients $\boldsymbol{\alpha} = [\alpha_1^1, \ldots, \alpha_1^L, \ldots, \alpha_k^1, \ldots, \alpha_k^L] \in \mathbb{R}^{Lk}$. Thus, this step is similar to eliciting linear metrics (Chapters 3,4) in the presence of an oracle which provides absolute quality feedback.

We therefore probe the metric $\widehat{\mathcal{E}}^{\text{val}}$ at Lm different classifiers $h^{1,1}, \ldots, h^{1,k}, \ldots, h^{L,1}, \ldots, h^{L,k}$, which results in a set of Lk linear equations of the form in (7.11):

$$\begin{aligned} \textstyle\sum_{\ell,i} \alpha_i^\ell\, \widehat{\Phi}_i^{\text{tr},\ell}[h^{1,1}] &= \widehat{\mathcal{E}}^{\text{val}}[h^{1,1}], \\ &\vdots \\ \textstyle\sum_{\ell,i} \alpha_i^\ell\, \widehat{\Phi}_i^{\text{tr},\ell}[h^{L,k}] &= \widehat{\mathcal{E}}^{\text{val}}[h^{L,m}], \end{aligned} \tag{7.12}$$

where $\widehat{\Phi}_i^{\text{tr},\ell}[h] = \frac{1}{n^{\text{tr}}}\sum_{(x,y)\in S^{\text{tr}}} \phi^\ell(x)\,\mathbf{1}(y = i)h_i(x)$ is evaluated on the training sample and the metric $\widehat{\mathcal{E}}^{\text{val}}[h] = \sum_i \beta_i\, \widehat{C}_{ii}^{\text{val}}[h]$ is evaluated on the validation sample.

More formally, let $\widehat{\mathbf{\Sigma}} \in \mathbb{R}^{Lk\times Lk}$ and $\widehat{\boldsymbol{\mathcal{E}}} \in \mathbb{R}^{Lk}$ denote the left-hand and right-hand side observations in (7.12), i.e.,:

$$\begin{aligned} \widehat{\Sigma}_{(\ell,i),(\ell',i')} &= \frac{1}{n^{\text{tr}}}\sum_{(x,y)\in S^{\text{tr}}} \phi^{\ell'}(x)\mathbf{1}(y = i')h_{i'}^{\ell,i}(x), \\ \widehat{\mathcal{E}}_{(\ell,i)} &= \widehat{\mathcal{E}}^{\text{val}}[h^{\ell,i}]. \end{aligned} \tag{7.13}$$

Then the weight coefficients are given by $\widehat{\boldsymbol{\alpha}} = \widehat{\mathbf{\Sigma}}^{-1}\widehat{\boldsymbol{\mathcal{E}}}$.

Algorithm 7.1 : ElicitWeights for Diagonal Linear Metrics

```
Input: $\widehat{\mathcal{E}}^{\text{val}}$, Basis functions $\phi^1, \ldots, \phi^L : \mathcal{X} \to [0,1]$, Training set $S^{\text{tr}} \sim \mu$, Val. set $S^{\text{val}} \sim D$, $\bar{h}$, $\epsilon$, $\mathcal{H}$, $\gamma, \omega$
If fixed classifier:
    Choose $h^{\ell,i}(x) = \epsilon\phi^{\ell}(x)\, e^{i}(x) + (1 - \epsilon\phi^{\ell}(x))\, \bar{h}(x)$
Else:
    $\bar{\mathcal{H}} = \{\tau h + (1-\tau)\bar{h} \,|\, h \in \mathcal{H}, \tau \in [0, \epsilon]\}$
    Pick $h^{\ell,i} \in \bar{\mathcal{H}}$ to satisfy (7.14) with slack $\gamma, \omega, \forall(\ell, i)$
Compute $\widehat{\mathbf{\Sigma}}$ and $\widehat{\boldsymbol{\mathcal{E}}}$ using (7.13) with metric $\widehat{\mathcal{E}}^{\text{val}}$
Output: $\widehat{\boldsymbol{\alpha}} = \widehat{\mathbf{\Sigma}}^{-1}\widehat{\boldsymbol{\mathcal{E}}}$
```

Algorithm 7.2 : Plug-**in** with **E**licited **W**eights (**PI-EW**) for Diagonal Linear Metrics

```
Input: $\widehat{\mathcal{E}}^{\text{val}}$, Basis functions $\phi^1, \ldots, \phi^L : \mathcal{X} \to [0,1]$, Class probability model $\widehat{\eta}^{\text{tr}} : \mathcal{X} \to \Delta_k$ for $\mu$, Training set $S^{\text{tr}} \sim \mu$, Validation set $S^{\text{val}} \sim D$, $\bar{h}$, $\epsilon$
$\widehat{\boldsymbol{\alpha}}$ = ElicitWeights($\widehat{\mathcal{E}}^{\text{val}}, \phi^1, \ldots, \phi^L, S^{\text{tr}}, S^{\text{val}}, \bar{h}, \epsilon$)
Example-weights: $\widehat{W}_i(x) = \sum_{\ell=1}^{L} \widehat{\alpha}_i^{\ell} \phi_i^{\ell}(x)$
Plug-in: $\widehat{h}(x) \in \operatorname{argmax}_{i \in [k]} \widehat{W}_i(x) \widehat{\eta}_i^{\text{tr}}(x)$
Output: $\widehat{h}$
```

7.3.4 Choosing the Probing Classifiers $h^{1,1}, \ldots, h^{L,k}$

We will have to choose the Lk probing classifiers so that $\widehat{\mathbf{\Sigma}}$ is well-conditioned. One way to do this is to choose the classifiers so that $\widehat{\mathbf{\Sigma}}$ has a high value on the diagonal entries and a low value on the off-diagonals, i.e. choose each classifier $h^{\ell,i}$ to evaluate to a high value on $\widehat{\Phi}_i^{\text{tr},\ell}[h]$ and a low value on $\widehat{\Phi}_{i'}^{\text{tr},\ell'}[h]$, $\forall\ (\ell', i') \neq (\ell, i)$. This can be framed as the following constraint satisfaction problem on S^{tr}:

For $h^{\ell,i}$ pick $h \in \mathcal{H}$ such that:

$$\widehat{\Phi}_i^{\text{tr},\ell}[h] \geq \gamma, \text{ and } \widehat{\Phi}_{i'}^{\text{tr},\ell'}[h] \leq \omega, \forall(\ell', i') \neq (\ell, i), \tag{7.14}$$

for some $\gamma > \omega > 0$ and a sufficiently flexible hypothesis class $\mathcal{H}$ for which the constraints are feasible. These problems can generally be solved by formulating a constrained classification problem [15, 105]. We show in Appendix E.7 that this problem is feasible and can be efficiently solved for a range of settings.

In practice, we do not explicitly solve (7.14) over a hypothesis class $\mathcal{H}$. Instead, a simpler and surprisingly effective strategy is to set the probing classifiers to *trivial* classifiers that predict the same class on all (or a subset of) examples. To build intuition for why this is a good idea, consider a simple setting with only one basis function $\phi^1(x) = 1, \forall x$, where the

ϕ-confusions $\widehat{\Phi}^{\mathrm{tr},1}_i[h] = \frac{1}{n^{\mathrm{tr}}}\sum_{(x,y)\in S^{\mathrm{tr}}} \mathbf{1}(y=i)h_i(x)$ are the standard confusion entries on the training set. In this case, a trivial classifier $e^i(x) = \mathrm{onehot}(i), \forall x$, which predicts class i on all examples, yields the highest value for $\widehat{\Phi}^{\mathrm{tr},1}_i$ and 0 for all other $\widehat{\Phi}^{\mathrm{tr},1}_j, \forall j \neq i$. In fact, in our experiments, we set the probing classifier $h^{1,i}$ to a randomized combination of e^i and some fixed base classifier $\bar{h}$:

$$h^{1,i}(x) = \epsilon e^i(x) + (1-\epsilon)\bar{h}(x), \tag{7.15}$$

for large enough ϵ so that $\widehat{\mathbf{\Sigma}}$ is well-conditioned.

Similarly, if the basis functions divide the data into L clusters (as in (7.8)), then we can randomize between $\bar{h}$ and a trivial classifier that predicts a particular class i on all examples assigned to the cluster $\ell \in [L]$. The confusion matrix for the resulting classifiers will have higher values than $\bar{h}$ on the (ℓ, i)-th diagonal entry and a lower value on other entries. These classifiers can be succinctly written as:

$$h^{\ell,i}(x) = \epsilon\phi^{\ell}(x)e^i(x) + (1-\epsilon\phi^{\ell}(x))\bar{h} \tag{7.16}$$

where we again tune ϵ to make sure that the resulting $\widehat{\mathbf{\Sigma}}$ is well-conditioned. This choice of the probing classifiers also works well in practice for general basis functions ϕ^{ℓ}'s.

Algorithm 7.1 summarizes the weight elicitation procedure, where the probing classifiers are either constructed by solving the constrained satisfaction problem (7.14) or set to the "fixed" classifiers in (7.16). In both cases, the algorithm takes a base classifier $\bar{h}$ and the parameter ϵ as input, where ϵ controls the extent to which $\bar{h}$ is perturbed to construct the probing classifiers. This radius parameter ϵ restricts the probing classifiers to a neighborhood around $\bar{h}$ and will prove handy in the algorithm we develop in Section 7.4.2.

7.4 PLUG-IN BASED ALGORITHMS

Having elicited the weight coefficients $\boldsymbol{\alpha}$, we now seek to learn a classifier that optimizes the left hand side of (7.11). We do this via the *plug-in* approach: first *pre-train* a model $\widehat{\eta}^{\mathrm{tr}} : \mathcal{X} \to \Delta_k$ on the noisy training distribution μ to estimate the conditional class probabilities $\widehat{\eta}^{\mathrm{tr}}_i(x) \approx \mathbf{P}^{\mu}(y=i|x)$, and then apply the correction weights to *post-shift* $\widehat{\eta}^{\mathrm{tr}}$.

7.4.1 Plug-in Algorithm for Linear Metrics

We first describe our approach for (diagonal) linear metrics $\mathcal{E}^D[h] = \sum_i \beta_i C^D_{ii}[h]$ in Algorithm 7.2. Given the correction weights $\widehat{\mathbf{W}} : \mathcal{X} \to \mathbb{R}^k_+$, we seek to maximize the following

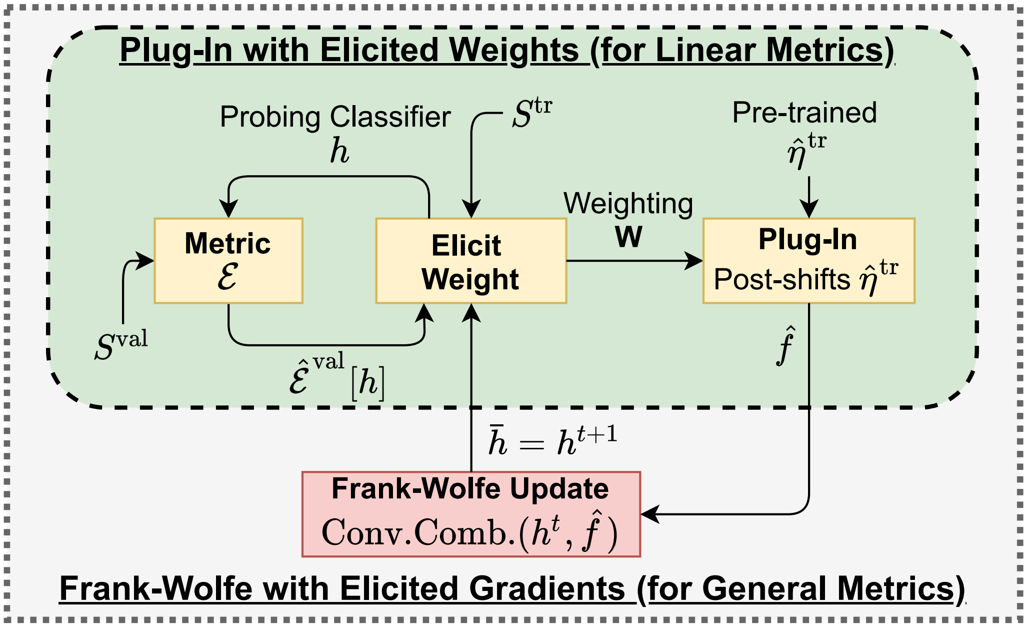

Plug-In with Elicited Weights (for Linear Metrics)
Probing Classifier
h
S^{tr}
Pre-trained
$\hat{\eta}^{\text{tr}}$
Weighting
$\mathbf{W}$
Metric
$\mathcal{E}$
Elicit Weight
Plug-In
Post-shifts $\hat{\eta}^{\text{tr}}$
S^{val}
$\hat{\mathcal{E}}^{\text{val}}[h]$
$\hat{f}$
$\bar{h} = h^{t+1}$
Frank-Wolfe Update
$\text{Conv.Comb.}(h^t, \hat{f})$
Frank-Wolfe with Elicited Gradients (for General Metrics)

Figure 7.1: Overview of our apporach.

weighted objective on the training distribution:

$$\max_h \, \mathbf{E}_{(x,y)\sim\mu}\left[\textstyle\sum_i \widehat{W}_i(x)\,\mathbf{1}(y=i)h_i(x)\right]. \tag{7.17}$$

This is a standard example-weighted learning problem, for which the following *plug-in* (also known as *post-shift*) classifier is a consistent estimator [18, 106]:

$$\widehat{h}(x) \in \operatorname*{argmax}_{i\in[k]} \widehat{W}_i(x)\,\widehat{\eta}_i^{\text{tr}}(x). \tag{7.18}$$

7.4.2 Iterative Algorithm for General Metrics

To optimize generic non-linear metrics of the form $\mathcal{E}^D[h] = \psi(C_{11}^D[h], \ldots, C_{kk}^D[h])$ for $\psi : [0,1]^k \to \mathbb{R}_+$, we apply Algorithm 7.2 iteratively. We consider both cases where ψ is unknown, and where ψ is known, but needs to be optimized using the noisy distribution μ. The idea is to first elicit local linear approximations to ψ and to then learn plug-in classifiers for the resulting linear metrics in each iteration.

Specifically, following Narasimhan et al.[18], we derive our algorithm from the classical Frank-Wolfe method [107] for maximizing a smooth concave function $\psi(\mathbf{c})$ over a convex set $\mathcal{C} \subseteq \mathbb{R}^m$. In our case, $\mathcal{C}$ is the set of confusion matrices $\mathbf{C}^D[h]$ achieved by any classifier h, and is convex when we allow randomized classifiers (see Lemma E.8, Appendix E.2.3). The

Algorithm 7.3 : **F**rank-**W**olfe with **E**licited **G**radients (**FW-EG**) for General Diagonal Metrics (also depicted in Fig. 7.1)

```
**Input:** $\widehat{\mathcal{E}}^{\text{val}}$, Basis functions $\phi^1, \ldots, \phi^L : \mathcal{X} \to [0,1]$, Pre-trained $\widehat{\eta}^{\text{tr}} : \mathcal{X} \to \Delta_k$, $S^{\text{tr}} \sim \mu$, $S^{\text{val}} \sim D$, $T$, $\epsilon$
Initialize classifier $h^0$ and $\mathbf{c}^0 = diag(\widehat{\mathbf{C}}^{\text{val}}[h^0])$
**For** $t = 0$ **to** $T - 1$ **do**
    **if** $\mathcal{E}^D[h] = \psi(C^D_{11}[h], \ldots, C^D_{kk}[h])$ for known $\psi$:
        $\boldsymbol{\beta}^t = \nabla\psi(\mathbf{c}^t)$
        $\widehat{\mathcal{E}}^{\text{lin}}[h] = \sum_i \beta^t_i \widehat{C}^{\text{val}}_{ii}[h]$
    **else**
        $\widehat{\mathcal{E}}^{\text{lin}}[h] = \widehat{\mathcal{E}}^{\text{val}}[h]$                {small $\epsilon$ recommendeded}
    $\widehat{f} = \textbf{PI-EW}(\widehat{\mathcal{E}}^{\text{lin}}, \phi^1, ..., \phi^L, \widehat{\eta}^{\text{tr}}, S^{\text{tr}}, S^{\text{val}}, h^t, \epsilon)$
    $\widetilde{\mathbf{c}} = diag(\widehat{\mathbf{C}}^{\text{val}}[\widehat{f}])$
    $h^{t+1} = \big(1 - \frac{2}{t+1}\big)h^t + \frac{2}{t+1}\text{onehot}(\widehat{f})$
    $\mathbf{c}^{t+1} = \big(1 - \frac{2}{t+1}\big)\mathbf{c}^t + \frac{2}{t+1}\widetilde{\mathbf{c}}$
**End For**
**Output:** $\widehat{h} = h^T$
```

algorithm maintains iterates $\mathbf{c}^t$, and at each step, maximizes a linear approximation to ψ at $\mathbf{c}^t$: $\widetilde{\mathbf{c}} \in \text{argmax}_{\mathbf{c} \in \mathcal{C}} \langle \nabla\psi(\mathbf{c}^t), \mathbf{c} \rangle$. The next iterate $\mathbf{c}^{t+1}$ is then a convex combination of $\mathbf{c}^t$ and the current solution $\widetilde{\mathbf{c}}$.

In Algorithm 7.3, we outline an adaptation of this Frank-Wolfe algorithm to our setting, where we maintain a classifier h^t and an estimate of the diagonal confusion entries $\mathbf{c}^t$ from the validation sample S^{val}. At each step, we linearize ψ using $\widehat{\mathcal{E}}^{\text{lin}}[h] = \sum_i \beta^t_i \widehat{C}^{\text{val}}_{ii}[h]$, where $\boldsymbol{\beta}^t = \nabla\psi(\mathbf{c}^t)$, and invoke the plug-in method in Algorithm 7.2 to optimize the linear approximation $\widehat{\mathcal{E}}^{\text{lin}}$. When the mathematical form of ψ is known, one can directly compute the gradient $\boldsymbol{\beta}^t$. When it is not known, we can simply set $\widehat{\mathcal{E}}^{\text{lin}}[h] = \widehat{\mathcal{E}}^{\text{val}}[h]$, but restrict the weight elicitation routine (Algorithm 7.1) to choose its probing classifiers $h^{\ell,i}$'s from a small neighborhood around the current classifier h^t (in which ψ is effectively linear). This can be done by passing $\bar{h} = h^t$ to the weight elicitation routine, and setting the radius ϵ to a small value.

Each call to Algorithm 7.2 uses the training and validation set to elicit example weights for a local linear approximation to ψ, and uses the weights to construct a plug-in classifier. The final output is a randomized combination of the plug-in classifiers from each step. Note that Algorithm 7.3 runs efficiently for reasonable values of L and k. Indeed the runtime is almost always dominated by the pre-training of the base model $\widehat{\eta}^{\text{tr}}$, with the time taken to elicit the weights (e.g. using (7.16)) being relatively inexpensive (see Appendix E.5).

7.5 THEORETICAL GUARANTEES

We provide theoretical guarantees for the weight elicitation procedure and the plug-in methods in Algorithms 7.1–7.3.

Assumption 7.1. The distributions D and μ are such that for any linear metric $\mathcal{E}^D[h] = \sum_i \beta_i C_{ii}[h]$, with $\|\boldsymbol{\beta}\| \leq 1$, $\exists \bar{\boldsymbol{\alpha}} \in \mathbb{R}^{Lk}$ s.t. $\left|\sum_{\ell,i} \bar{\alpha}_i^\ell \Phi_i^{\mu,\ell}[h] - \mathcal{E}^D[h]\right| \leq \nu, \forall h$ and $\|\bar{\boldsymbol{\alpha}}\|_1 \leq B$, for some $\nu \in [0,1)$ and $B > 0$.

The assumption states that our choice of basis functions $\phi^1, \ldots, \phi^L$ are such that, any linear metric on D can be approximated (up to a slack ν) by a weighting $W_i(x) = \sum_\ell \bar{\alpha}_i^\ell \phi^\ell(x)$ of the training examples from μ. The existence of such a weighting function depends on how well the basis functions capture the underlying distribution shift. Indeed, the assumption holds for some common settings in Table 7.1, e.g., when the noise transition $\mathbf{T}$ is diagonal (Appendix E.1 handles a general $\mathbf{T}$), and the basis functions are set to $\phi^1(x) = 1, \forall x$, for the IDLN setting, and $\phi^\ell(x) = \mathbf{1}(g(x) = \ell), \forall x$, for the CDLN setting.

We analyze the coefficients $\widehat{\boldsymbol{\alpha}}$ elicited by Algorithm 7.1 when the probing classifiers $h^{\ell,i}$ are chosen to satisfy (7.14). In Appendix E.3, we provide an analysis when the probing classifiers $h^{\ell,i}$ are set to the fixed choices in (7.16).

Theorem 7.1 (Error bound on elicited weights). Let $\gamma, \omega > 0$ be such that the constraints in (7.14) are feasible for hypothesis class $\bar{\mathcal{H}}$, for all ℓ, i. Suppose Algorithm 7.1 chooses each classifier $h^{\ell,i}$ to satisfy (7.14), with $\mathcal{E}^D[h^{\ell,i}] \in [c, 1], \forall \ell, i$, for some $c > 0$. Let $\bar{\alpha}$ be defined as in Assumption 7.1. Suppose $\gamma > 2\sqrt{2}Lm\omega$ and $n^{\text{tr}} \geq \frac{L^2 m \log(Lm|\mathcal{H}|/\delta)}{(\frac{\gamma}{2} - \sqrt{2}Lm\omega)^2}$. Fix $\delta \in (0,1)$. Then w.p. $\geq 1 - \delta$ over draws of S^{tr} and S^{val} from μ and D resp., the coefficients $\widehat{\boldsymbol{\alpha}}$ output by Algorithm 7.1 satisfies:

$$\|\widehat{\boldsymbol{\alpha}} - \bar{\boldsymbol{\alpha}}\| \leq \mathcal{O}\bigg(\frac{Lk}{\gamma^2}\bigg(\sqrt{\frac{L\log(\frac{Lk|\mathcal{H}|}{\delta})}{n^{\text{tr}}}} + \sqrt{\frac{L\log(\frac{Lk}{\delta})}{c^2 n^{\text{val}}}}\bigg) + \frac{\nu\sqrt{Lk}}{\gamma}\bigg), \tag{7.19}$$

where the term $|\mathcal{H}|$ can be replaced by a measure of capacity of the hypothesis class $\mathcal{H}$.

Because the probing classifiers are chosen using the training set alone, it is only the sampling errors from the training set that depend on the complexity of $\mathcal{H}$, and not those from the validation set. This suggests robustness of our approach to a small validation set as long as the training set is sufficiently large and the number of basis functions is reasonably small.

For the iterative plug-in method in Algorithm 7.3, we bound the gap between the metric value $\mathcal{E}^D[\widehat{h}]$ for the output classifier $\widehat{h}$ on the true distribution D, and the optimal value.

We handle the case where the function ψ is *known* and its gradient $\nabla\psi$ can be computed in closed-form. The more general case of an unknown ψ is handled in Appendix E.4. The above bound depends on the gap between the estimated class probabilities $\widehat{\eta}_i^{\text{tr}}(x)$ for the training distribution and true class probabilities $\eta_i^{\text{tr}}(x) = \mathbf{P}(y = i|x)$, as well as the quality of the coefficients $\widehat{\boldsymbol{\alpha}}$ provided by the weight estimation subroutine, as measured by $\kappa(\cdot)$. One can substitute $\kappa(\cdot)$ with, e.g., the error bound provided in Theorem 7.1.

Theorem 7.2 (Error Bound for FW-EG). Let $\mathcal{E}^D[h] = \psi(C_{11}^D[h], \ldots, C_{kk}^D[h])$ for a *known* concave function $\psi : [0,1]^k \to \mathbb{R}_+$, which is Q-Lipschitz and λ-smooth. Fix $\delta \in (0,1)$. Suppose Assumption 7.1 holds, and for any linear metric $\sum_i \beta_i C_{ii}^D[h]$, whose associated weight coefficients is $\bar{\boldsymbol{\alpha}}$ with $\|\bar{\boldsymbol{\alpha}}\| \leq B$, w.p. $\geq 1-\delta$ over draw of S^{tr} and S^{val}, the weight estimation routine in Alg. 7.1 outputs coefficients $\widehat{\boldsymbol{\alpha}}$ with $\|\widehat{\boldsymbol{\alpha}} - \bar{\boldsymbol{\alpha}}\| \leq \kappa(\delta, n^{\text{tr}}, n^{\text{val}})$, for some function $\kappa(\cdot) > 0$. Let $B' = B + \sqrt{Lk}\,\kappa(\delta/T, n^{\text{tr}}, n^{\text{val}})$. Then w.p. $\geq 1-\delta$ over draws of S^{tr} and S^{val} from D and μ resp., the classifier $\widehat{h}$ output by Algorithm 7.3 after T iterations satisfies:

$$\max_h \mathcal{E}^D[h] - \mathcal{E}^D[\widehat{h}] \;\leq\; 2QB'\mathbf{E}_x\left[\|\eta^{\text{tr}}(x) - \widehat{\eta}^{\text{tr}}(x)\|_1\right] + 4Q\sqrt{Lk}\,\kappa(\tfrac{\delta}{T}, n^{\text{tr}}, n^{\text{val}}) + \mathcal{O}\left(\lambda k\sqrt{\frac{k\log(k)\log(n^{\text{val}}) + \log(k/\delta)}{n^{\text{val}}}} + \frac{\lambda}{T} + Q\nu\right). \tag{7.20}$$

The proof in turn derives an error bound for the plug-in classifier in Algorithm 7.2 for linear metrics (see Appendix E.2.2).

7.6 RELATED WORK

Methods for closed-form metrics. There has been a variety of work on optimizing complex evaluation metrics, including both plug-in type algorithms [8, 18, 108, 109, 110], and those that use convex surrogates for the metric [61, 111, 112, 113, 114, 115, 116]. These methods rely on the test metric having a specific closed-form structure and do not handle black-box metrics.

Methods for black-box metrics. Among recent black-box metric learning works, the closest to ours is by Jiang et al.[91], who learn a weighted combination of surrogate losses to approximate the metric on a validation set. Like us, they probe the metric at multiple classifiers, but their approach has several drawbacks on both practical and theoretical fronts. Firstly, Jiang et al. [91] require retraining the model in each iteration, which can be time-intensive, whereas we only post-shift a pre-trained model. Secondly, the procedure they prescribe for eliciting gradients requires perturbing the model parameters multiple times, which can be very expensive for large deep networks, whereas we only require perturbing the predictions from the model. Moreover, the number of perturbations they need grows *polynomially* with the precision with which they need to estimate the loss coefficients, whereas we only require a *constant* number of them. Lastly, their approach does not come with strong statistical guarantees, whereas ours does. Besides these benefits over [91], we will also see in Section 7.7 that our method yields better accuracies. Other related black-box learning methods include [90, 95, 96], who learn a (weighted) loss to approximate the metric, but do so using computationally expensive procedures (e.g. meta-gradient descent or RL) that often require retraining the model from scratch, and come with limited theoretical analysis.

Methods for distribution shift. The literature on distribution shift is vast, and so we cover a few representative papers; see [117, 118] for a comprehensive discussion. For the *independent label noise* setting [93], Patrini et al. [94] propose a loss correction approach that first trains a model with noisy label, use its predictions to estimate the noise transition matrix, and then re-trains model with the corrected loss. This approach is however tailored to optimize linear metrics; whereas, we can handle more complex metrics as well without re-training the underlying model. A plethora of approaches exist for tackling *domain shift*, including classical importance weighting (IW) strategies [92, 119, 120, 121] that work in two steps: estimate the density ratios and train a model with the resulting weighted loss. One such approach is Kernel Mean Matching [122], which matches covariate distributions between training and test sets in a high dimensional RKHS feature space. These IW approaches are however prone to over-fitting when used with deep networks [123]. More recent iterative variants seek to remedy this [124].

7.7 EXPERIMENTS

We run experiments on four classification tasks, with both known and black-box metrics, and under different label noise and domain shift settings. All our experiments use a large training sample, which is either noisy or contains missing attributes, and a smaller clean (and complete) validation sample. We always optimize the cross-entropy loss for learning $\widehat{\eta}^{\mathrm{tr}}(x) \approx \mathbf{P}^{\mu}(Y|x)$ using the training set (or $\widehat{\eta}^{\mathrm{val}}(x) \approx \mathbf{P}^{D}(Y|x)$ for some baselines), where the models are varied across experiments. For monitoring the quality of $\widehat{\eta}^{\mathrm{tr}}$ and $\widehat{\eta}^{\mathrm{val}}$, we sample small subsets *hyper-train* and *hyper-val* data from the original training and validation data, respectively. We repeat our experiments over 5 random train-vali-test splits, and report the mean and standard deviation for each metric. We will use *, **, and *** to denote that the differences between our method and the closest baseline are statistically significant (using Welch's t-test) at a confidence level of 90%, 95%, and 99%, respectively. We provide the

Table 7.2: Data Statistics for different problem setups in Section 7.7.

Problem Setup	**Dataset**	**#Classes**	**#Features**	**train / val / test split**
Indepen. Label Noise (Section 7.7.1)	CIFAR-10	10	$32 \times 32 \times 3$	49K / 1K / 10K
Proxy-Label (Section 7.7.2)	Adult	2	101	32K / 350 / 16K
Domain-Shift (Section 7.7.3)	Adience	2	$256 \times 256 \times 3$	12K / 800 / 3K
Black-Box Fairness Metric (Section 7.7.4)	Adult	2 (2 prot. groups)	106	32K / 1.5K / 14K

data statistics in Table 7.2. Observe that we always use small validation data in comparison to the size of the training data. The source code (along with random seeds) is provided on the link below.[1]

Common baselines: We use representative baselines from the black-box learning [91], iterative re-weighting [95], label noise correction [94], and importance weighting [122] literatures. First, we list the ones common to all experiments.

1. **Cross-entropy [train]:** Maximizes accuracy on the training set and predicts:

$$\widehat{h}(x) \in \operatorname*{argmax}_{i\in[k]} \widehat{\eta}_i^{\text{tr}}(x). \tag{7.21}$$

2. **Cross-entropy [val]:** Maximizes accuracy on the validation set and predicts:

$$\widehat{h}(x) \in \operatorname*{argmax}_{i\in[k]} \widehat{\eta}_i^{\text{val}}(x). \tag{7.22}$$

3. **Fine-tuning:** Fine-tunes the pre-trained $\widehat{\eta}^{\text{tr}}$ using the validation data, monitoring the cross-entropy loss on the hyper-val data for early stopping.

4. **Opt-metric [val]:** For metrics $\psi(\mathbf{C}^D[h])$, for which ψ is *known*, trains a model to directly maximize the metric on the small *validation* set using the Frank-Wolfe based algorithm of [18].

5. **Learn-to-reweight** [95]: Jointly learns example weights, with the model, to maximize accuracy on the validation set; does not handle specialized metrics.

6. **Plug-in [train-val]:** Constructs a classifier $\widehat{h}(x) \in \operatorname{argmax}_i w_i\widehat{\eta}_i^{\text{val}}(x)$, where the weights $w_i \in \mathbb{R}$ are tuned to maximize the given metric on the validation set, using a coordinate-wise line search (details in Appendix E.6).

7. **Adaptive Surrogates** [91]: Learns a weighted combination of surrogate losses (evaluated on clusters of examples) to approximate the metric on the validation set. Since

[1] `https://github.com/koyejolab/fweg/`

this method is not directly amenable for use with large neural networks (see Section 7.6), we compare with it only when using linear models, and present additional comparisons in App. E.8 (Table E.1).

Hyper-parameters: The learning rate for Fine-tuning is chosen from $1\mathrm{e}^{\{-6,\ldots,-4\}}$. For PI-EW and FW-EG, we tune the parameter ϵ from $\{1, 0.4, 1\mathrm{e}^{-\{4,3,2,1\}}\}$. The line search for Plug-in is performed with a spacing of $1\mathrm{e}^{-4}$. The only hyper-parameters the other baselines have are those for training $\widehat{\eta}^{\mathrm{tr}}$ and $\widehat{\eta}^{\mathrm{val}}$, which we state in the individual tasks.

7.7.1 Maximizing Accuracy under Label Noise

In our first task, we train a 10-class image classifier for the CIFAR-10 dataset [97], replicating the independent (asymmetric) label noise setup from [94]. The evaluation metric we use is accuracy. We take 2% of original training data as validation data and flip labels in the remaining training set based on the following transition matrix: TRUCK → AUTOMOBILE, BIRD → PLANE, DEER → HORSE, CAT ↔ DOG, with a flip probability of 0.6. For $\widehat{\eta}^{\mathrm{tr}}$ and $\widehat{\eta}^{\mathrm{val}}$, we use the same ResNet-14 architecture as [94], trained using SGD with momentum 0.9, weight decay $1\mathrm{e}^{-4}$, and learning rate 0.01, which we divide by 10 after 40 and 80 epochs (120 in total).

We additionally compare with the *Forward Correction* method of [94], a specialized method for correcting independent label noise, which estimates the noise transition matrix $\mathbf{T}$ using predictions from $\widehat{\eta}^{\mathrm{tr}}$ on the training set, and retrains it with the corrected loss, thus training the ResNet twice. We saw a notable drop with this method when we used the (small) validation set to estimate $\mathbf{T}$.

We apply the proposed PI-EW method for linear metrics, using a weighting function $\mathbf{W}$ defined with one of two choices for the basis functions (chosen via cross-validation): (i) a default basis function that clusters all the points together $\phi^{\mathrm{def}}(x) = 1\,\forall x$, and (ii) ten basis functions $\phi^1, \ldots, \phi^{10}$, each one being the average of the RBF kernels (see (7.9)) centered at validation points belonging to a true class. The RBF kernels are computed with width 2 on UMAP-reduced 50-dimensional image embeddings [125].

As shown in Table 7.3, PI-EW achieves significantly better test accuracies than all the baselines. The results for Forward Correction matches those in [94]; unlike this method, we train the ResNet only once, but achieve 2.4% higher accuracy. Cross-entropy [val] over-fits badly, and yields the least test accuracy. Surprisingly, the simple fine-tuning yields the second-best accuracy. A possible reason is that the pre-trained model learns a good feature representation, and the fine-tuning step adapts well to the domain change. We also

Table 7.3: Test accuracy for noisy label experiment on CIFAR-10.

Cross-entropy [train]	0.582 ± 0.007
Cross-entropy [val]	0.386 ± 0.031
Learn-to-reweight	0.651 ± 0.017
Plug-in [train-val]	0.733 ± 0.044
Forward Correction	0.757 ± 0.005
Fine-tuning	0.769 ± 0.005
PI-EW	**0.781 ± 0.019**

observed that PI-EW achieves better accuracy during cross-validation with ten basis functions, highlighting the benefit of the underlying modeling in PI-EW. Lastly, in Figure 7.2(a), we show the elicited (class) weights with the default basis function ($\phi^{\text{def}}(x) = 1\,\forall x$), where e.g. because BIRD → PLANE, the weight on BIRD is upweighted and that on PLANE is down-weighted.

7.7.2 Maximizing G-mean with Proxy Labels

Our next experiment borrows the "proxy label" setup from [91] on the Adult dataset [99]. The task is to predict whether a candidate's gender is male, but the training set contains only a proxy for the true label. We sample 1% validation data from the original training data, and replace the labels in the remaining sample with the feature 'relationship-husband'. The label noise here is instance-dependent (see Example 7.3), and we seek to maximize the G-mean metric:

$$\psi(\mathbf{C}) \,=\, \Big(\prod_i \big(C_{ii}/\sum_j C_{ij}\big)\Big)^{1/m}. \tag{7.23}$$

We train $\widehat{\eta}^{\text{tr}}$ and $\widehat{\eta}^{\text{val}}$ using linear logistic regression using SGD with a learning rate of 0.01. As additional baselines, we include the Adaptive Surrogates method of [91] and *Forward Correction* [94]. The inner and outer learning rates for Adaptive Surrogates are each cross-validated in $\{0.1, 1.0\}$. We also compare with a simple Importance Weighting strategy, where we first train a logistic regression model f to predict if an example (x, y) belongs to the validation data, and train a gender classifier with the training examples weighted by $f(x, y)/(1 - f(x, y))$.

We choose between three sets of basis functions (using cross-validation): (i) a default basis function $\phi^{\text{def}}(x) = 1\,\forall x$, (ii) $\phi^{\text{def}}, \phi^{\text{pw}}, \phi^{\text{npw}}$, where $\phi^{\text{pw}}(x) = \mathbf{1}(x_{\text{pw}} = 1)$ and $\phi^{\text{npw}}(x) = \mathbf{1}(x_{\text{npw}} = 1)$ use features 'private-workforce' and 'non-private-workforce' to form hard clusters, (iii) ϕ^{def}, $\phi^{\text{pw}}, \phi^{\text{npw}}, \phi^{\text{inc}}$, where $\phi^{\text{inc}}(x) = \mathbf{1}(x_{\text{inc}} = 1)$ uses the binary feature 'income'. These choices are motivated from those used by [91], who compute surrogate losses on the

Table 7.4: Test G-mean for proxy label experiment on Adult.

Cross-entropy [train]	0.654 ± 0.002
Cross-entropy [val]	0.394 ± 0.064
Opt-metric [val]	0.652 ± 0.027
Learn-to-reweight	0.668 ± 0.003
Plug-in [train-val]	0.672 ± 0.013
Forward Correction	0.214 ± 0.004
Fine-tuning	0.631 ± 0.017
Importance Weights	0.662 ± 0.024
Adaptive Surrogates	0.682 ± 0.002
FW-EG [unknown ψ]	$\mathbf{0.685 \pm 0.002}^{**}$
FW-EG [known ψ]	$\mathbf{0.685 \pm 0.001}^{*}$

individual clusters. We provide their Adaptive Surrogates method with the same clustering choices.

Table 7.4 summarizes our results. We apply both variants of our FW-EG method for a non-linear metric ψ, one where ψ is *known* and its gradient is available in closed-form, and the other where ψ is assumed to be *unknown*, and is treated as a general black-box metric. Both variants perform similarly and are better than the baselines. Adaptive Surrogates comes a close second, but underperforms by 0.3% (with results being statistically significant). While the improvement of FW-EG over Adaptive Surrogates is small, the latter is time intensive as, in each iteration, it re-trains a logistic regression model. We verify this empirically in Figure 7.2(b) by reporting run-times for Adaptive Surrogates and our method FW-EG (including the pre-training time) against the choices of basis functions (clustering features). We see that our approach is 5× faster for this experiment. Lastly, Forward Correction performs poorly, likely because its loss correction is not aligned with this label noise model.

7.7.3 Maximizing F-measure under Domain Shift

We now move on to a domain shift application (see Example 7.4). The task is to learn a gender recognizer for the Adience face image dataset [98], but with the training and test datasets containing images from different age groups (domain shift based on age). We use images belonging to age buckets 1–5 for training (12.2K images), and evaluate on images from age buckets 6–8 (4K images). For the validation set, we sample 20% of the 6–8 age bucket images. Here we aim to maximize the F-measure.

For $\widehat{\eta}^{\mathrm{tr}}$ and $\widehat{\eta}^{\mathrm{val}}$, we use the same ResNet-14 model from the CIFAR-10 experiment, except that the learning rate is divided by 2 after 10 epochs (20 in total). As an additional baseline, we compute importance weights using *Kernel Mean Matching (KMM)* [122], and train the

Table 7.5: Test F-measure for domain shift experiment on Adience.

Cross-entropy [train]	0.760 ± 0.014
Cross-entropy [val]	0.708 ± 0.022
Opt-metric [val]	0.760 ± 0.014
Plug-in [train-val]	0.759 ± 0.014
Importance Weights [KMM]	0.760 ± 0.013
Learn-to-reweight	0.773 ± 0.009
Fine-tuning	0.781 ± 0.014
FW-EG [unknown ψ]	$\mathbf{0.815 \pm 0.013}^{***}$
FW-EG [known ψ]	$\mathbf{0.804 \pm 0.015}^{***}$

same ResNet model with a weighted loss. Since the image size is large for directly applying KMM, we first compute the 2048-dimensional ImageNet embedding [126] for the images and further reduce them to 10-dimensions via UMAP. The KMM weights are learned on the 10-dimensional embedding. For the basis functions, besides the default basis $\phi^{\mathrm{def}}(x) = 1\, \forall x$, we choose from subsets of six RBF basis functions $\phi^1, \ldots, \phi^6$, centered at points from the validation set, each representing one of six age-gender combinations. We use the same UMAP embedding as KMM to compute the RBF kernels.

Table 7.5 presents the test F-measure values. Both variants of FW-EG algorithm provide statistically significant improvements over the baselines. Both Fine-tuning and Learning-to-reweight improve over plain cross-entropy optimization (train), however only moderately, likely because of the small size of the validation set, and because these methods are not tailored to optimize the F-measure.

7.7.4 Maximizing Black-box Fairness Metric

We next handle a black-box metric given only query access to its value. We consider a fairness application where the goal is to balance classification performance across multiple protected groups. The groups that one cares about are known, but due to privacy or legal restrictions, the protected attribute for an individual cannot be revealed [89]. Instead, we have access to an oracle that reveals the value of the fairness metric for predictions on a validation sample, with the protected attributes absent from the training sample. This setup is different from recent work on learning fair classifiers from incomplete group information [46, 74], in that the focus here is on optimizing *any* given black-box fairness metric.

We use the Adult dataset, and seek to predict whether the candidate's income is greater than $50K, with *gender* as the protected group. The black-box metric we consider (whose form is unknown to the learner) is the geometric mean of the true-positive (TP) and true-

Table 7.6: Black-box fairness metric on the test set for Adult.

Cross-entropy [train]	0.736 ± 0.005
Cross-entropy [val]	0.610 ± 0.020
Learn-to-reweight	0.729 ± 0.007
Fine-tuning	0.738 ± 0.005
Adaptive Surrogates	0.812 ± 0.004
Plug-in [train-val]	0.812 ± 0.005
FW-EG	$\mathbf{0.822 \pm 0.002}^{***}$

negative (TN) rates, evaluated separately on the male and female examples, which promotes equal performance for both groups and classes:

$$\mathcal{E}^D[h] = \Big(\mathrm{TP}^{\mathrm{male}}[h]\,\mathrm{TN}^{\mathrm{male}}[h])\,\mathrm{TP}^{\mathrm{female}}[h]\,\mathrm{TN}^{\mathrm{female}}[h]\Big)^{1/4}. \tag{7.24}$$

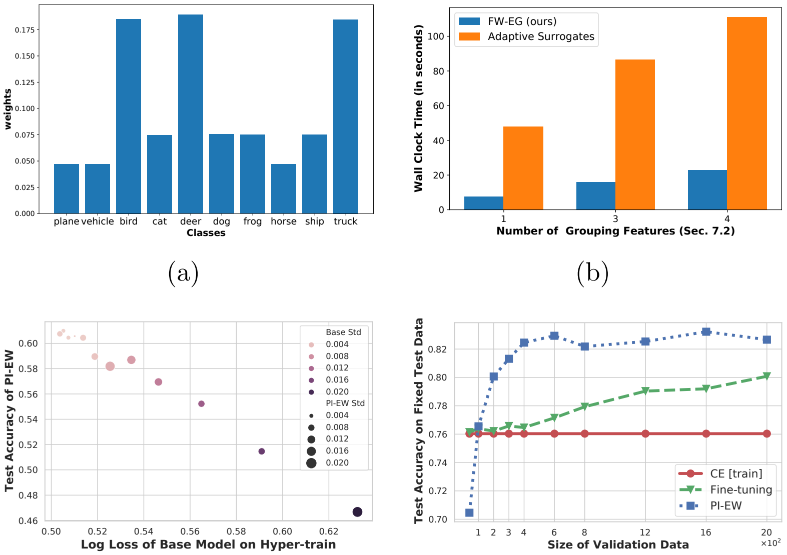
We train the same logistic regression models as in previous Adult experiment in Section 7.7.2. Along with the basis functions ϕ^{def}, ϕ^{pw} and ϕ^{npw} we used there, we additionally include two basis ϕ^{hs} and ϕ^{wf} based on features 'relationship-husband' and 'relationship-wife', which we expect to have correlations with gender.[2] We include two baselines that can handle black-box metrics: Plug-in [train-val], which tunes a threshold on $\widehat{\eta}^{\mathrm{tr}}$ by querying the metric on the validation set, and Adaptive Surrogates. The latter is cross-validated on the same set of clustering features (i.e., basis functions in our method) for computing the surrogate losses.

As seen in Table 7.6, FW-EG yields the highest black-box metric on the test set, Adaptive Surrogates comes in second, and surprisingly the simple plug-in approach fairs better than the other baselines. During cross-validation, we also observed that the performance of FW-EG improves with more basis functions, particularly with the ones that are better correlated with gender. Specifically, FW-EG with basis functions $\{\phi^{\mathrm{def}}, \phi^{\mathrm{pw}}, \phi^{\mathrm{npw}}, \phi^{\mathrm{wf}}, \phi^{\mathrm{hs}}\}$ achieves approximately 1% better performance than both FW-EG with ϕ^{def} basis function and FW-EG with basis functions $\{\phi^{\mathrm{def}}, \phi^{\mathrm{pw}}, \phi^{\mathrm{npw}}\}$.

7.7.5 Ablation Studies

We close with two sets of experiments. First, we analyze how the performance of PI-EW, while optimizing accuracy for the Adult experiment (Section 7.7.2), varies with the quality of the base model $\widehat{\eta}^{\mathrm{tr}}$. We save an estimate of $\widehat{\eta}^{\mathrm{tr}}$ after every 50 batches (batch size 32) while

[2] The only domain knowledge we use is that the protected group is "gender"; beyond this, the form of the metric is unknown, and importantly, an individual's gender is not available.



(a)

(b)

(c)

(d)

Figure 7.2: (a) Elicited (class) weights for CIFAR-10 by PI-EW for the default basis (Sec. 7.7.1); (b) Run-time for FW-EG and Adaptive Surrogates [91] vs no. of grouping features on proxy label task (Sec. 7.7.2); (c) Effect of quality of the base model $\widehat{\eta}^{\text{tr}}$ on Adult (Sec. 7.7.2): as the base model's quality improves, the test accuracies of PI-EW also improves; (d) Effect of the validation set size on Adience (Sec. 7.7.3): PI-EW performs better than fine-tuning even for small validation sets, while both improve with larger ones.

training the logistic regression model, and use these estimates as inputs to PI-EW. As shown in Figure 7.2(c), the test accuracies for PI-EW improves with the quality of $\widehat{\eta}^{\text{tr}}$ (as measured by the log loss on the hyper-train set). This is in accordance with Theorem 7.2. One can further improve the quality of the estimate η^{tr} by using calibration techniques [127], which will likely enhance the performance of PI-EW as well.

Next, we show that PI-EW is robust to changes in the validation set size when trained on the Adience experiment in Section 7.7.3 to optimize accuracy. We set aside 50% of 6–8 age bucket data for testing, and sample varying sizes of validation data from the rest. As shown in Figure 7.2(d), PI-EW generally performs better than fine-tuning even for small validation sets, while both improve with larger ones. The only exception is 100-sized validation set (0.8% of training data), where we see overfitting due to small validation size.

7.7.6 Black-box optimization with pairwise comparison oracle

The proposed algorithm in this chapter works with *machine* oracles that when queried for a classifier h respond with the metric value $\mathcal{E}^D[h]$. We saw various cases, e.g., validation set in distribution shift settings or a regulator in fairness setups, where we have access to such an oracle. We exploit the fact that the example weights act as a gradient or a local linear objective in a small neighborhood for the unknown metric, and elicit such linear metrics through the use of value queries to the machine oracle.

The same idea can be extended in the presence of a *human* oracle that provides pairwise preferences. This also includes A/B testing scenarios commonly used in the web based applications. In order to elicit a local-linear objective around a classifier's confusion matrix $C^D[h]$, one can first construct a small sphere around $C^D[h]$ and the corresponding classifiers by the process discussed in Section B.4.1, and then run Algorithm 4.2 to elicit the local-linear performance metric using the pairwise comparisons. Once the local-linear objective is estimated, then one can post-shift a pre-trained class-conditional estimator similar to the proposed FW-EG algorithm (Algorithm 7.3).

However, this approach comes with its own challenges. Firstly, in order to apply the iterative Frank-Wolfe approach, one will need to create the spheres and the corresponding classifiers multiple times. This would make the algorithm time-intensive as it would require to solve an optimization problem in each iteration. Secondly, it is not clear how to elicit the local-linear objective, when one chooses overlapping or softly clustered basis functions. We hope to overcome these challenges in the future.

7.8 CONCLUDING REMARKS

In this chapter, we proposed the Frank Wolfe with Elicited Gradient (FW-EG) method for optimizing black-box metrics given query access to the evaluation metric on a small validation set. Our framework includes common distribution shift settings as special cases, and unlike prior distribution correction strategies, is able to handle general non-linear metrics. A key benefit of our method is that it is agnostic to the choice of $\widehat{\eta}^{\text{tr}}$, and can thus be used to post-shift pre-trained deep networks, without having to retrain them. We showed that the post-shift example weights can be flexibly modeled with various choices of basis functions (e.g., hard clusters, RBF kernels, etc.) and empirically demonstrated their efficacies. We exploit the fact that the example weights act as a gradient for the unknown metric and estimated through metric elicitation procedure, where a *machine* oracle responds with absolute quality value of a classifier on a clean validation dataset. Moreover, the novel geometrical characterizations discussed in Chapters 3, 4, and 5 led us to devise an efficient and a smart method for creating the probing classifiers (see (7.16)). We look forward to further improving the results with more nuanced basis functions.

CHAPTER 8: PRACTICAL METRIC ELICITATION

Till now, our contributions towards the Metric Elicitation (ME) framework with pairwise comparisons have been algorithmic. So, to bring theory closer to practice, in this chapter, we conduct a preliminary real-user study that shows the efficacy of the metric elicitation framework in recovering the users' preferred performance metrics in a binary classification setup.

We choose *cancer diagnosis* [53] as the application for this task, where the ground-truth label is a binary feature denoting whether or not the patient has cancer. This choice is motivated by Application 1 discussed in Chapter 1, since there are asymmetric costs associated with False Positives and False Negatives – based on known consequences of misdiagnosis, i.e, side-effects of treating a healthy patient vs. mortality rate for not treating a sick patient. Our work (a) builds upon existing visualizations for confusion matrices to ask for pairwise preferences, and (b) then try to elicit a linear performance metric using our proposed procedure in Algorithm 3.1 in the binary classification setup. We work with ten subjects in this preliminary study, who have some experience either with machine learning or biomedical research in the university setup.

We create a web User Interface (UI),[1] which broadly has three parts to it. First, it shows subjects a couple of confusion matrices and asks questions related to *comprehension, comparison, and simulation* [128]. These questions familiarize the subjects with the visualizations and the components associated with the correct and incorrect predictions. Second, it shows a bunch of pairwise preference queries over confusion matrices. The UI involves running the binary-search procedure from Algorithm 3.1 at the back end, which chooses the next set of queries based on the subject's current real-time responses. Third, the UI comprises of fifteen pairwise comparison queries, where the confusion matrices are randomly chosen from the feasible set. The responses to these queries are used to evaluate the fidelity of the recovered metric through metric elicitation framework. At the end of the web-based task, the subjects are asked some subjective questions which essentially lead to our guidelines that we recommend for implementing the metric elicitation framework in real-life scenarios.

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. In addition, the goal includes testing certain assumptions regarding the noise in the subject's (oracle's) responses, work around with finite samples, and

[1] The user-interface is shown later and is also available at http://safinahali.com/elicitation-graphs-static/

provide future guidance on visualizing confusion matrices for pairwise comparisons, eliciting actual performance metrics in real-life scenarios, and evaluating the quality of the recovered metric.

The contributions from this chapter are summarized as follows:

- We create a web UI that uses existing visualizations of confusion matrices that are refined to capture preferences over pairwise comparisons.
- The UI implements the binary-search procedure from Algorithm 3.1 at the back end that make use of the real-time responses over confusion matrices to elicit a linear performance metric in the cancer diagnosis setup.
- 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. The study also includes a post-task, *think-aloud*-style interview regarding the utility of the framework.
- Lastly, using the task results and the post-task interviews, we present guidelines regarding practical implementation of the ME framework that can be used for future research in this direction.

8.1 DATASET AND VISUALIZATION CHOICE

In this section, we first discuss the details of the dataset used and how the feasible set of confusion matrices is constructed. Then, we discuss the choice of visualizations for confusion matrices, which are borrowed from prior work, but are refined to allow for better pairwise comparisons.

8.1.1 Choice of Task and Dataset Used

Our choice of task domain and the dataset is motivated by Application 1 discussed in Chapter 1. The task is *cancer diagnosis* [53] for which we use the Breast Cancer Wisconsin (Original) dataset from the UCI repository.[2] 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.

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

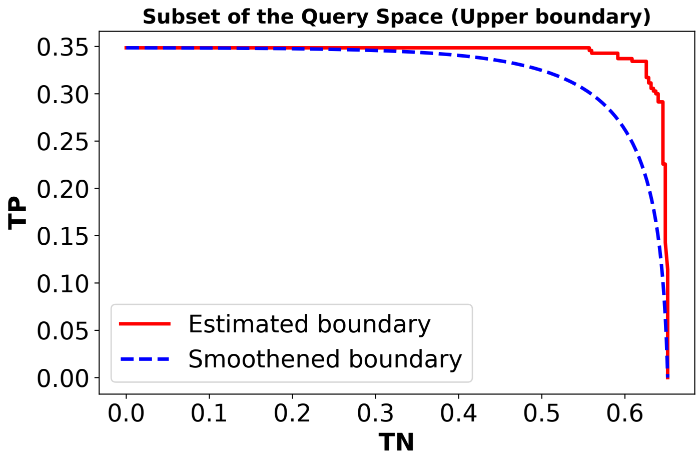


Figure 8.1: Estimated confusions on test data forming the upper boundary of the space of confusion matrices and the associated smoothened version of the upper boundary.

Around 35% of the data is labelled as 1 and the rest as 0. 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., $\widehat{\eta}(x) = \widehat{\mathbb{P}}(y = 1|X)$. We then create a pool of thresholded classifiers of the type:

$$h_\tau(x) = \mathbf{1}[\widehat{\eta}(x) \geq \tau], \tag{8.1}$$

where we vary the threshold τ from 0 to 1 in steps of $1e^{-4}$. Subsequently, we compute confusion matrices for the above threhsolded classifiers on the test data (resulting in 10001 confusion matrices). As discussed in Chapter 3 (see Figure 3.1), the space of confusion matrices is a two-dimensional space and the confusions (tuple of true positives and true negatives) associated with the thresholed classifiers above form the upper boundary. This upper boundary for the estimated confusions on the test data is shown in Figure 8.1 (see solid, red line).

As discussed in Chapter 3, one can use these estimated confusion matrices in practice to elicit linear performance metrics. However, in the binary classification setup, we can easily smoothen the upper boundary, and that too using feasible confusion matrices. This allows to reduce the *staircase* type bumps due to estimation from finite data, and consequentially, lead to better convergence from the binary-search based Algorithm 3.1. To generate confusions on the smoothened version of the upper boundary, we take the same simulated distribution setting from Section 3.6.1.

Specifically, we take a joint probability for $\mathcal{X} = [-1, 1]$ and $\mathcal{Y} = \{0, 1\}$ given by $f_X =$

$\mathbb{U}[-1,1]$ and $\eta(x) = \frac{1}{1+e^{ax+b}}$, where $\mathbb{U}[-1,1]$ is the uniform distribution on $[-1,1]$. Then we estimate the parameters a and b such that they minimize the squared error between the (10K) confusions obtained on the test data and the ones simulated by using the above distribution. The smoothened upper boundary is shown as dashed, blue curve in Figure 8.1. Clearly, all these confusions are feasible as they would lie inside the region enclosed by the upper and lower boundary, and thus we can use the confusions on the smoothened upper boundary for elicitation purposes.

8.1.2 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. [128]. 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. [128] are created over multiple iterative user-studies.

In the first study, the authors conduct interviews with 7 subjects and a survey with 102 subjects and map out two major sets of challenges lay people have in understanding standard confusion matrices. These are (a) general terminologies and (b) the matrix design. These challenges are further elaborated with three sub-challenges that include confusion about the direction of reading the data, layered relations, and the quantities involved. In order to tackle these challenges, the authors came up with four alternative visualizations of the confusion matrix. In the second study, the authors evaluate the efficacy of the proposed visualizations over 483 subjects on a recidivism prediction task [128]. The authors conclude that the *flow-chart* is the most preferred visualization of a confusion matrix followed by a *bar-chart*. Both these visualizations are shown in Figure 8.2 in the context of a recidivism prediction task.

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 [129], we choose to use the top two performing visualizations, i.e., the *flow-chart* and the *bar-chart*, together to depict a confusion matrix.

2. We transform the data statistics so that the numbers denote out-of-100 samples.

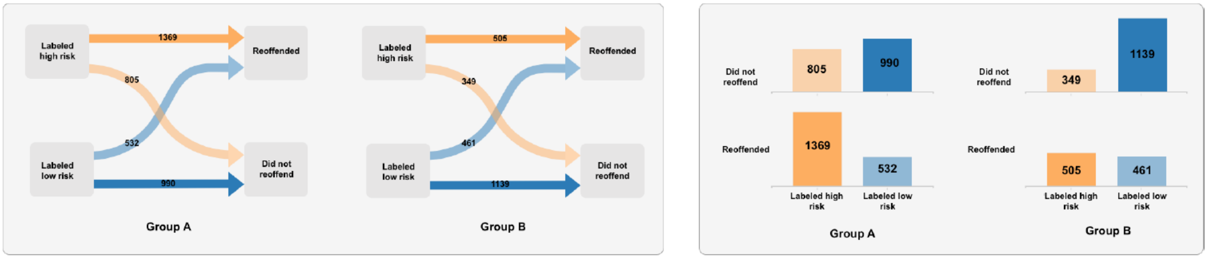


Figure 8.2: Flow-chart and bar-chart based visualizations for (binary classification) confusion matrices in the recidivism prediction task from Shen et al. [128].

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.

4. We also add a zoom-in feature for both the graphs for better understanding.

5. Although, in this preliminary user study, we have not changed the direction in the flow-chart, in our discussions with HCI and machine learning researchers, we also noted that the current direction is perhaps more important for the recidivism task (that is because there is time component involved with it) but can be changed for the cancer diagnosis task. This allows one to have constants (i.e., total positive and negative labels) in the left column and the varying component (i.e., total positive and negative predictions) on the right column making the comparison easier. Moreover, this change ensures that the bar-chart and the flow-chart represent similar information. We plan to implement this change and record its impact in our future user studies.

Our modified visualization incorporating the first four points above for a confusion matrix in the context of cancer diagnosis is shown in Figure 8.3. We next discuss the web user interface.

8.2 USER INTERFACE

We discuss our proposed web User Interface (UI) in detail and discuss our rationale behind its several components. We also provide images of the UI at the end of this chapter.

The UI starts with a questionnaire asking about demographic information like age, gender, race, highest level of school, and the subjects' expertise in machine learning and healthcare as shown in Figure 8.4. Then the UI has three parts to it as explained in the following sub-sections.

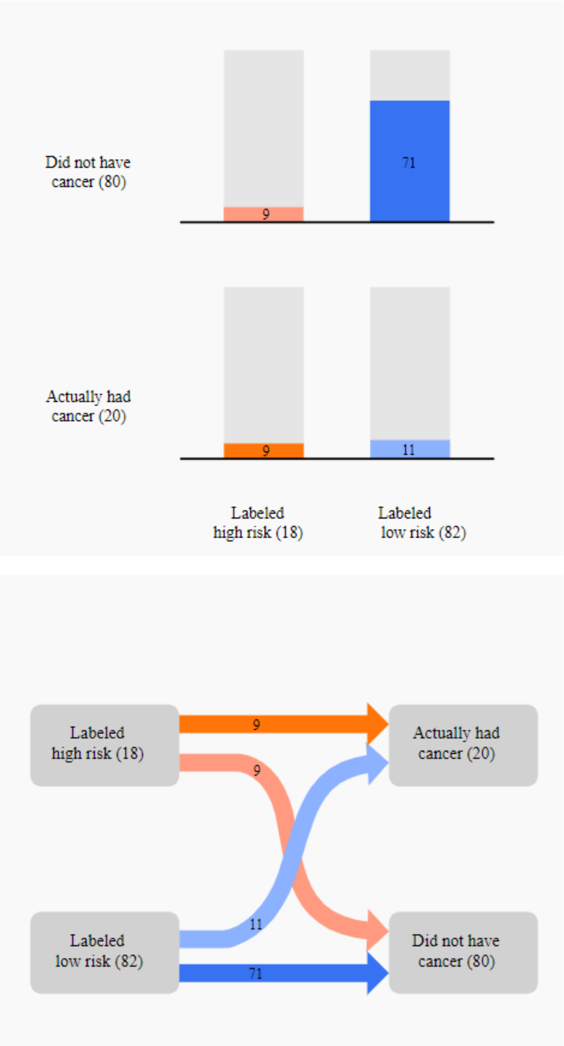


Figure 8.3: Our modified visualization of a confusion matrix for a cancer diagnosis task. Modification is from the perspective of obtaining better pairwise preferences.

8.2.1 Understanding and Familiarizing with the Visualizations

After the questionnaire, we describe the task of cancer diagnosis and provide details on how classifiers can be inaccurate in their predictions in layman terms. We also show the proposed visualization of a confusion matrix along with the description as exhibited in Figure 8.5.

On the next four pages, we show visualizations of two confusion matrices side by side and ask a series of questions regarding the data depicted in them. The first three adapt the questions from Shen et al. [128] for the cancer diagnosis task. See Figures 8.6-8.8 for the UI snapshots. Shen et al. [128] framed these questions to evaluate the *comprehension*, *comparison*[3], and *simulation*-based understanding of the subjects. We use these questions to

[3] The comparison questions in Shen et al. [128] are different than pairwise comparisons like ours. They

make them familiarize with the visualizations. The fourth page asks the subjects to actually compare two hypothetically created confusion matrices (see Figure 8.9). Here, one of the matrices has both higher false positives and false negatives. This question has a definitive answer and was added to make the subjects familiarize with the type of pairwise comparison questions that would follow. In addition, this question indicates how good the subject has grasped the context around cancer diagnosis and the task of pairwise comparisons.

8.2.2 Practically Eliciting Linear Performance Metrics

We next explain the second phase of the UI, where we actually ask subjects for pairwise preferences over confusion matrices, and implement our binary-search procedure from Algorithm 3.1. The confusion matrices used for this procedure are from the smoothened upper boundary shown in Figure 8.1; thus, the subjects have to make a choice reflecting on the trade-off between false positives and false negatives. Algorithm 3.1 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 3 of Algorithm 3.1). Moreover, in practice, we do not need to ask four queries per round of binary search; instead, we can reduce the search interval into half by just using at most three pairwise queries in each round (i.e., by querying $\Omega(\overline{C}_{\theta_c}, \overline{C}_{\theta_a}), \Omega(\overline{C}_{\theta_d}, \overline{C}_{\theta_c}), \Omega(\overline{C}_{\theta_e}, \overline{C}_{\theta_d})$, in line 6 of Algorithm 3.1). A sample of a pairwise comparison query from a run of the binary search algorithm in the UI is shown in Figure 8.10.

8.2.3 Pairwise Preferences on a Random Set of Queries

In order to evaluate the quality of the recovered metric, 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. The confusion matrices are generated from a sphere of radius 0.1 around the center (0.35/2, 0.65/2). This set of queries are used to evaluate the effectiveness of the elicited metric. We compute the fraction of times our elicited metric's preferences matches with the subject's preferences on these fifteen queries. A sample of a pairwise comparison query from this phase of the UI is shown in Figure 8.11. We ask fifteen such queries.

focus on comparing just one component, e.g., true positives, at a time.

Table 8.1: Subjects' demographics: Distribution of responses from the questionnaire. The values in parenthesis show the number of subjects.

Age	25 (2)	26 (3)	28 (5)
Education Level	in Graduate College (4)	Master's (3)	Doctorate (3)
ML Expertise	None (5)	Beginner (3)	Intermediate (2)
Healthcare Knowledge	None (5)	Some (2)	No response (3)

Table 8.2: Post-task interview questions.

Q1	What do you think is worse: (a) Large number of patients that actually have cancer but are labelled as low risk by a computer system, or (b) Large number of patients that do not have cancer but are labelled as high risk by a computer system.
Q2	Could you quantify how much worse the chosen option is in comparison to the other? Why or why not? Could you quantify this personally? i.e, 10x worse for me
Q3	For the questions presented in this task, how did you decide which system you would prefer your doctor to use?
Q4	What was difficult about making these choices?
Q5	What additional information would have helped you to make these choices?
Q6	Do you have any feedback for us on your experience today?

8.3 USER STUDY

We hired ten subjects in total for this preliminary study. The study was conducted over a video call, where the participants were asked to share the screen after they had filled the questionnaire on the first page. The distributions of the responses from the questionnaire are provided in Table 8.1. The rest of the responses regarding the confusion matrices were over screen share and were logged in the UI. After the task was done, the web UI showed a 'thank you' page and asked the subjects to close the web browser and screen share. The subjects were then asked post-task, *think-aloud* interview questions, which are shown in Table 8.2, to reflect on how they performed the given task. The responses from the interviews help us formulate guidelines and recommendations for future research in this direction.

8.4 RESULTS

In this section, we discuss results from the preliminary user-study both quantitatively and qualitatively. We will try to answer some of the practical questions that surround the metric elicitation framework as discussed in the beginning of this chapter. Specifically, we focus on checking workflow of the practical implementation, support or reject the hypothesis that the implicit user preferences can be quantified using the pairwise comparison queries,

Table 8.3: Summary of guidelines and recommendations from the user-study.

G1	Whenever possible, smoothen the query space so to run the binary-search based algorithms with reduced finite sample errors.
G2	Depending on the search tolerance of the binary-search, show probabilities in the confusion matrix as out-of-n samples, where bigger the n, the better it is to differentiate between confusion matrices in a query.
G3	The direction in the flow-chart based visualization of the confusion matrix can be swapped with total number of labels shown in the left column and total predictions on the right.
G4	Perhaps, showing only flow-chart for pairwise comparisons is better than showing flow-chart and bar-chart together. One may also just show, the false positives and false negatives to further reduce the information load.
G5	Measure time to respond for each query. Spending more time on queries that comprise close confusion matrices lead credence to the noise model in Definition 2.4.
G6	The terminology "labelled as high risk/low risk" can be replaced with "predicted as high risk/low risk" to avoid confusions regarding ground-truth label.
G7	In view of the post-interview question number 2, one needs to devise a UI so to ask for the intuitive guess for the false negative cost. This would also act as a baseline metric for evaluation purposes (see Section 8.4.1).
G8	One can also have a toggle button that shows percentages conditioned on the true classes (i.e., in addition to false positive and false negative, one can have false positive rate and false negative rate). This would aid in making comparisons.
G9	Extend the description on cancer diagnosis and mention the associated (subjective) cost or excerpts that cover different aspects of the cost. For example, how much financial burden a false positive prediction would put on a patient, how much emotional burden would it put, what are the possible side-effects of drugs, etc.

testing assumptions regarding the noise model, work around with finite samples, visualizing confusion matrices for pairwise comparisons, eliciting actual performance metrics in real-life scenarios, and evaluating the quality of the recovered metric. We emphasize that the aim behind discussing results from the user study is to formulate guidelines and recommendations for future research on practical metric elicitation. We provide these recommendations as we discuss quantitative and qualitative results and summarize them in Table 8.3.

8.4.1 Quantitative Results and Findings

Impact of Smoothened Query Space and Out-of-100 Samples: We first discuss the impact of smoothening of the upper boundary from Section 8.1.1. Since we choose to ask pairwise preferences over confusion matrices directly, and not over classifiers, we provided a way to generate feasible confusion matrices in Section 8.1.1 that lie on the smoothened

Table 8.4: 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

version of the upper boundary. As we discussed in Section 3.6.2, working with finite samples has a drawback that the elicitation routine can get stuck at the closest achievable confusion matrix from finite samples, which need not be optimal within the given (small) tolerance. We find that working with the smoothened version almost always avoids asking pairs that comprise same confusion matrices, and thus guaranteeing better convergence within the chosen binary-search tolerance. We also note that showing probabilities in the form of out-of-10000 or bigger samples instead of out-of-samples 100 allows us to further reduce the cases where the confusion matrices are same in a pair or the comparisons becomes trivial (e.g., same false negatives but different false positives) for the subjects.

Elicited Metrics and Quality Evaluation: We next discuss the metrics that were elicited for the ten subjects using our web UI, which runs the binary-search based procedure Algorithm 3.1 at the back end. Once the search interval is less than or equal to 0.05, the subjects were asked fifteen queries that we use for evaluation. The measure of effectiveness that we choose is the fraction of times (in %) our elicited metric's preferences matches with the subject's preferences over the fifteen queries, i.e.,

$$\mathcal{M} := \frac{\sum_{i=1}^{15} \mathbf{1}[\text{subject's prefer. for query } i == \text{metric's prefer. for query } i]}{15} \times 100. \quad (8.2)$$

We show the elicited metric for the fifteen subjects and the measure $\mathcal{M}$ values in Table 8.4. 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.

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 $\mathcal{M}$.

8.4.2 Qualitative Feedback

We first describe the general feedback that was observed and discussed with the subjects during the user study over the video sessions. We formulate some guidelines from this feedback. We then mention a few excerpts from the post-task interviews again formulating some recommendations for practical metric elicitation.

Observations during Study Sessions: Similar to the observation by Shen et al. [128], in our user study as well, we also noted that subjects were not very comfortable with answering the *simulation*-based questions (see Figure 8.8). A possible reason is that the direction of the flow-chart is opposite to the conditioning of probability that is asked in those questions. Bar-chart allows them to answer this question easily; however, we find that by this point in the UI, the subject becomes more comfortable with using the flow-chart. Some users when asked in the post-interview session also mentioned that this could help them better in the pairwise comparison, too.

While comparing confusion matrices in the UI, we observed that after a few rounds, the subjects tend to look at only the flow-charts for comparison. This may mean showing the bar-charts and flow-charts together is overwhelming, and perhaps only the flow-charts are enough. After a few more rounds, some subjects started comparing only flow of false positives and false negatives in the flow-chart. This suggests that one may further reduce the information load by showing only false positives and false negatives in the flow chart.

Although, we do not quantitatively measure *time to respond* in this version of the UI, but we did observe that the subjects tend to take more time while comparing two confusion matrices that are close (i.e., the queries in the later part of the binary search when the search interval is narrow). This means that the subjects are more prone to make errors for such queries, leading credence to the noise model in Definition 2.4 that is used in this manuscript throughout.

Lastly, during the study, we found that some subjects, who were familiar with machine learning, confused the terminology "labelled as high risk/low risk" for predictions to the ground-truth labels. One suggestion is to replace the word "labelled" with "predicted".

Post-task Interview Sessions: We now discuss post-task interviews and formulate some guidelines. We also mention some excerpts (anonymously) from the interviews. Please see Table 8.2 for the interview questions.

Q1. Every subject clearly figured out the direction of the costs and mentioned that (in the words of S1), *"a patient who has cancer but was predicted as low risk is a costlier mistake than a patient who does not have cancer but was predicted as high risk."*

Q2. None of the subjects could answer this question with full confidence. This acts as a testimony to the importance of the metric elicitation framework. Often, practitioners make a guess to quantify the asymmetric costs in class-imbalanced learning; however, the guess may be far from innate costs of the practitioner. The subjects agreed that it is easier to compare two confusion matrices using the proposed visualizations than to answer this question.

Q3. Most of the subjects mention that they preferred the one where false negatives were less. Although some subjects looked at the trade-off, for example, (in the words of S2) *"I was trying to minimize the false negatives but not when very large number of false positives were there."* This reflects that some subjects had to think hard about the trade-offs.

Q4. The subjects mention that deciding on the trade-offs between false positives and false negatives was difficult. (In words of S6) *"It was difficult to pick a preference where both false positives and false negatives needed to be compared"*. Some subjects also mentioned that, (in words of S4), *"In some cases, numbers are really close; thus, it becomes difficult to select one of them"*. This feedback certainly agrees with the choice of the noise model in this manuscript (see Definition 2.4).

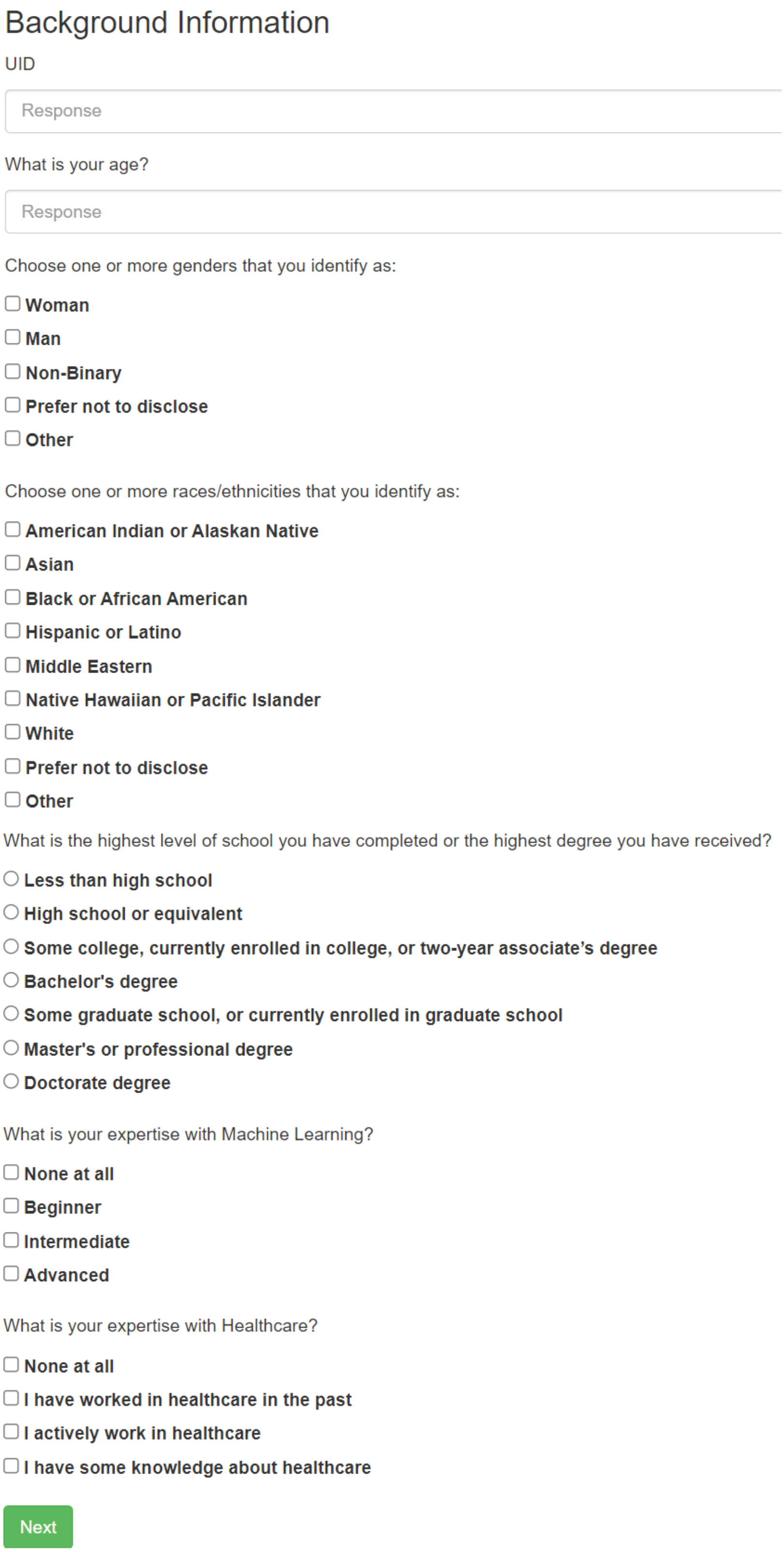
Q5. The responses to this question were important for constructing the guidelines, and this question had varied responses. One subject mentioned that having false positive rate and false negative rate, in addition to false positives and false negatives, would be helpful in making comparisons. (In words of S1), *"One can have percentages on the arrow conditioned on the samples in the box from which they are flowing."* Similarly, some subjects mentioned that it would have been easier to compare if the stages of cancer were mentioned in the predictions; the different stages would have lead to difference preferences. Some subjects quote that some description of the associated costs or excerpts that cover different aspects of the cost, at least subjectively should be described in the beginning of the study. For example, (in words of S2), *"how much financial burden a false positive prediction would put on a patient, how much emotional burden would it put, what are the possible side-effects of drugs, etc. should be highlighted in the beginning."*

Q6. Most subjects enjoyed the exercise and liked the web UI. Some subjects mentioned that the task allowed them to reflect closely on some important questions regarding performance metrics in machine learning.

8.5 CONCLUDING REMARKS

We created a web user-interface (UI) to practically elicit (linear) performance metrics with real users in a binary classification setup. We chose cancer diagnosis as the task domain, because it involves asymmetric costs for false positives and false negatives. We build upon existing visualizations of confusion matrices that are refined to capture preferences over pairwise comparisons. Via this user-study, we demonstrated an implementation of the binary performance metric elicitation procedure from Chapter 3 that make use of the real-time user responses over pairwise comparisons of confusion matrices. We also proposed and implemented an evaluation scheme to judge the quality of the recovered metric.

Using the proposed web UI, we then conducted a preliminary user study with ten subjects and elicited their linear performance metrics. We also compared 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. The study also included a post-task, *think-aloud*-style interviews regarding the utility of the framework. Using the task results and the feedback during the post-task interviews, we presented guidelines and recommendations for practical implementation of the ME framework. In the future, we plan to build upon this pilot study and conduct a comprehensive user study that includes the guidelines presented in this chapter with more subjects. We also plan to extend the current web UI to elicit metrics in the multiclass classification setup.



Background Information

UID

Response

What is your age?

Response

Choose one or more genders that you identify as:

□ **Woman**

□ **Man**

□ **Non-Binary**

□ **Prefer not to disclose**

□ **Other**

Choose one or more races/ethnicities that you identify as:

□ **American Indian or Alaskan Native**

□ **Asian**

□ **Black or African American**

□ **Hispanic or Latino**

□ **Middle Eastern**

□ **Native Hawaiian or Pacific Islander**

□ **White**

□ **Prefer not to disclose**

□ **Other**

What is the highest level of school you have completed or the highest degree you have received?

○ **Less than high school**

○ **High school or equivalent**

○ **Some college, currently enrolled in college, or two-year associate’s degree**

○ **Bachelor's degree**

○ **Some graduate school, or currently enrolled in graduate school**

○ **Master's or professional degree**

○ **Doctorate degree**

What is your expertise with Machine Learning?

□ **None at all**

□ **Beginner**

□ **Intermediate**

□ **Advanced**

What is your expertise with Healthcare?

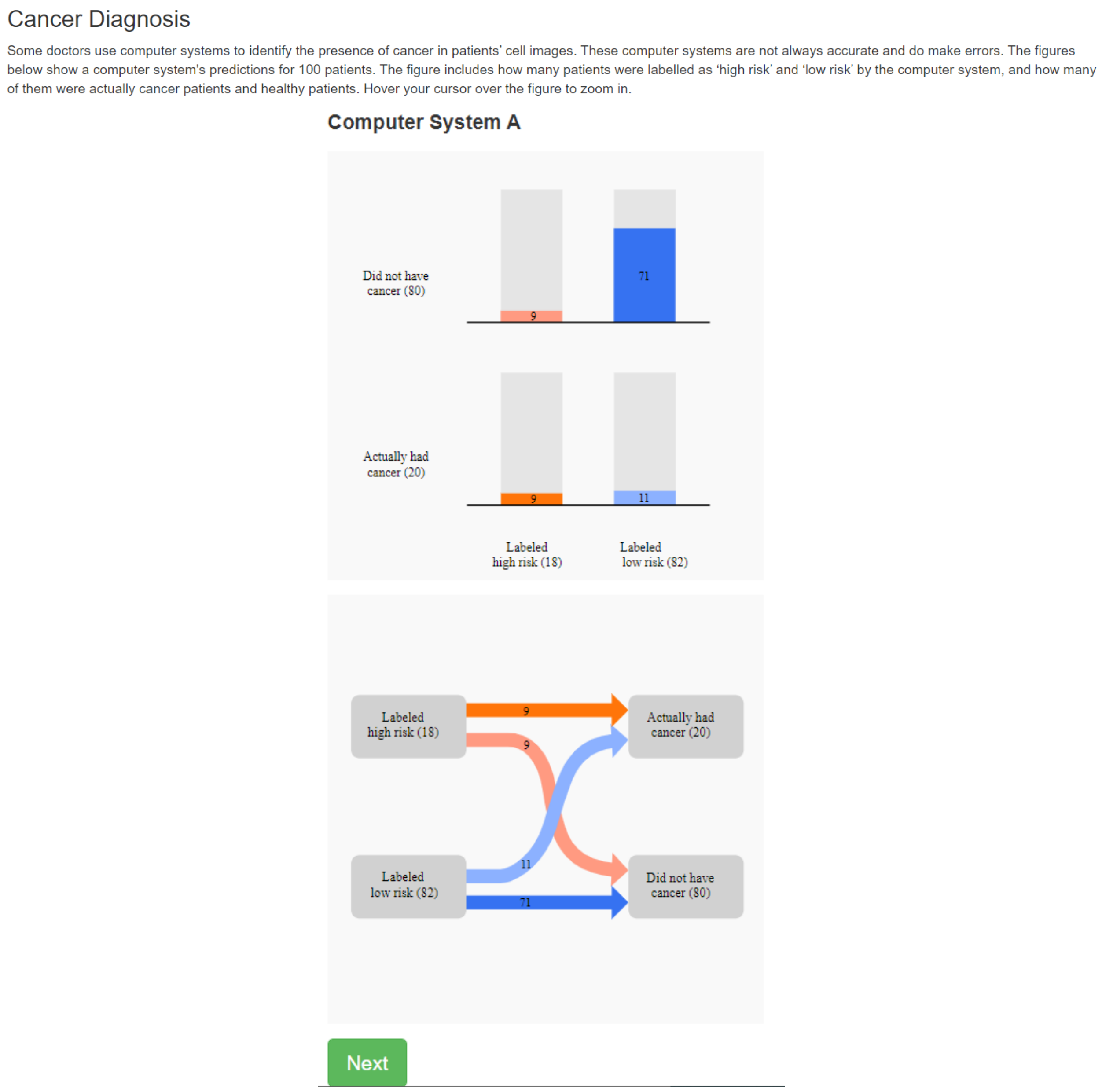
□ **None at all**

□ **I have worked in healthcare in the past**

□ **I actively work in healthcare**

□ **I have some knowledge about healthcare**

Figure 8.4: Questionnaire on the first page of the UI.

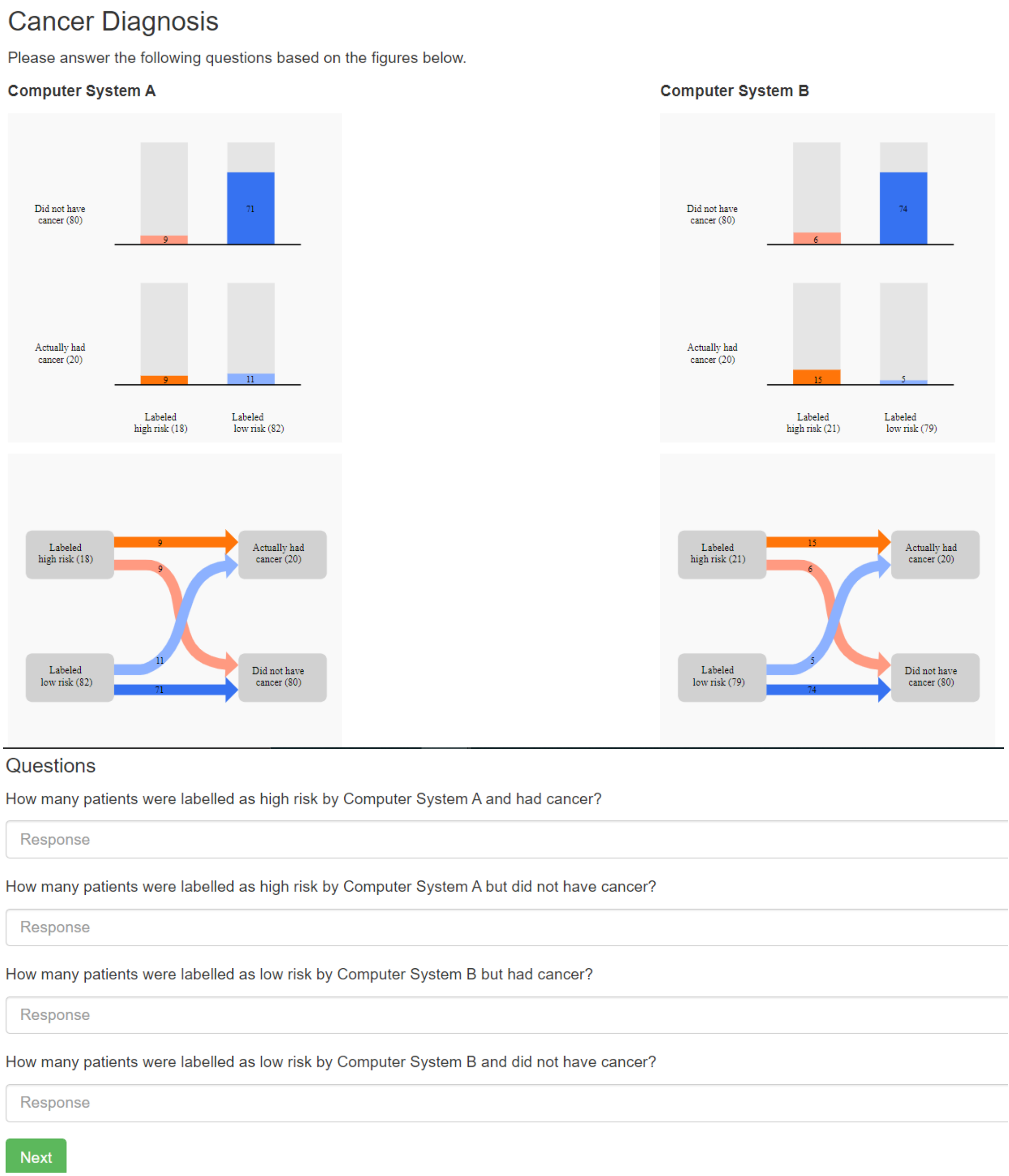


Cancer Diagnosis

Some doctors use computer systems to identify the presence of cancer in patients' cell images. These computer systems are not always accurate and do make errors. The figures below show a computer system's predictions for 100 patients. The figure includes how many patients were labelled as 'high risk' and 'low risk' by the computer system, and how many of them were actually cancer patients and healthy patients. Hover your cursor over the figure to zoom in.

Computer System A

Figure 8.5: Description of cancer diagnosis along with visualization of a confusion matrix.



Cancer Diagnosis

Please answer the following questions based on the figures below.

Computer System A

Computer System B

Questions

How many patients were labelled as high risk by Computer System A and had cancer?

Response

How many patients were labelled as high risk by Computer System A but did not have cancer?

Response

How many patients were labelled as low risk by Computer System B but had cancer?

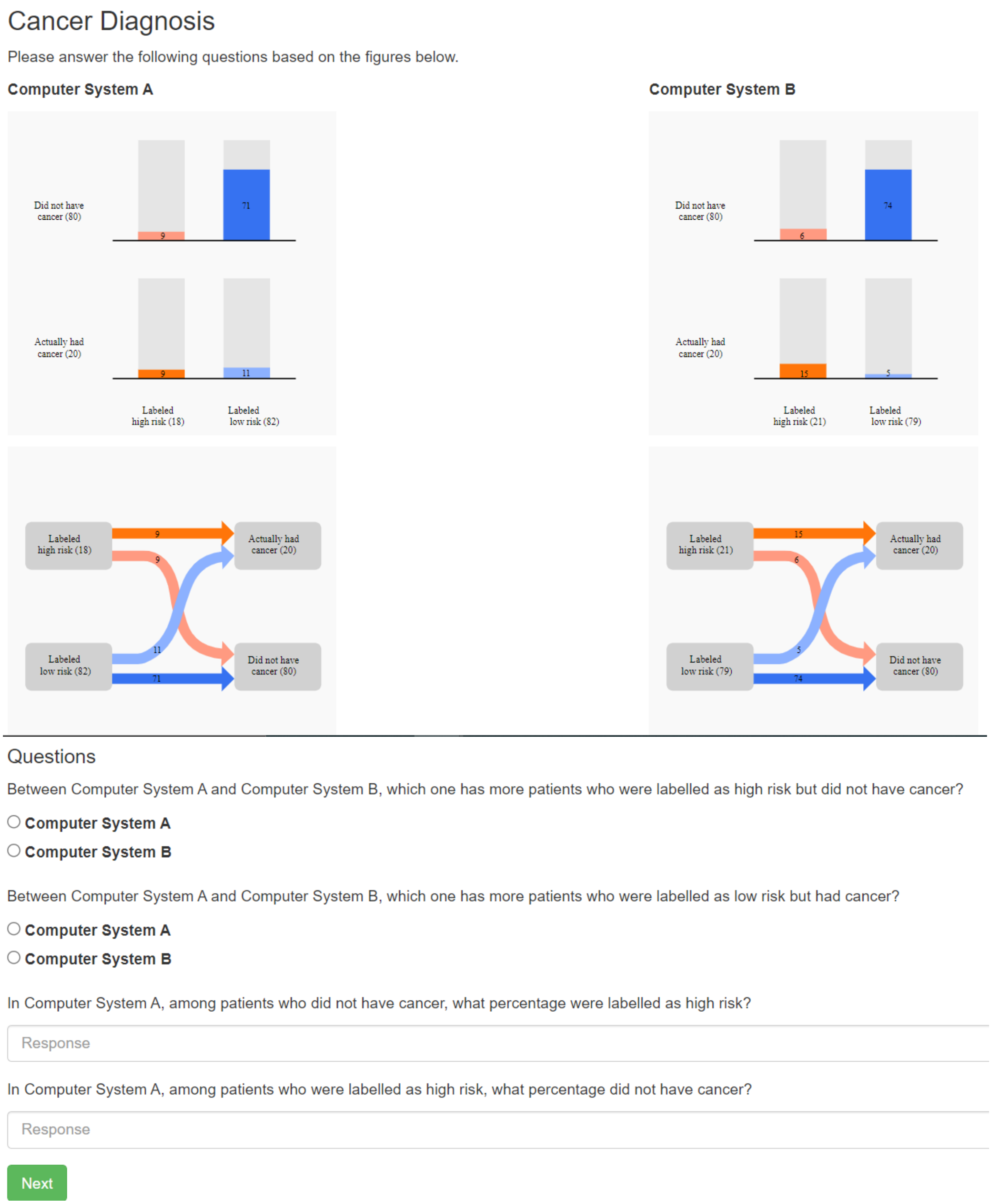
Response

How many patients were labelled as low risk by Computer System B and did not have cancer?

Response

Next

Figure 8.6: Two confusion matrices side by side. This page asks questions about *comprehending* confusion matrices.



Cancer Diagnosis

Please answer the following questions based on the figures below.

Questions

Between Computer System A and Computer System B, which one has more patients who were labelled as high risk but did not have cancer?

○ **Computer System A**

○ **Computer System B**

Between Computer System A and Computer System B, which one has more patients who were labelled as low risk but had cancer?

○ **Computer System A**

○ **Computer System B**

In Computer System A, among patients who did not have cancer, what percentage were labelled as high risk?

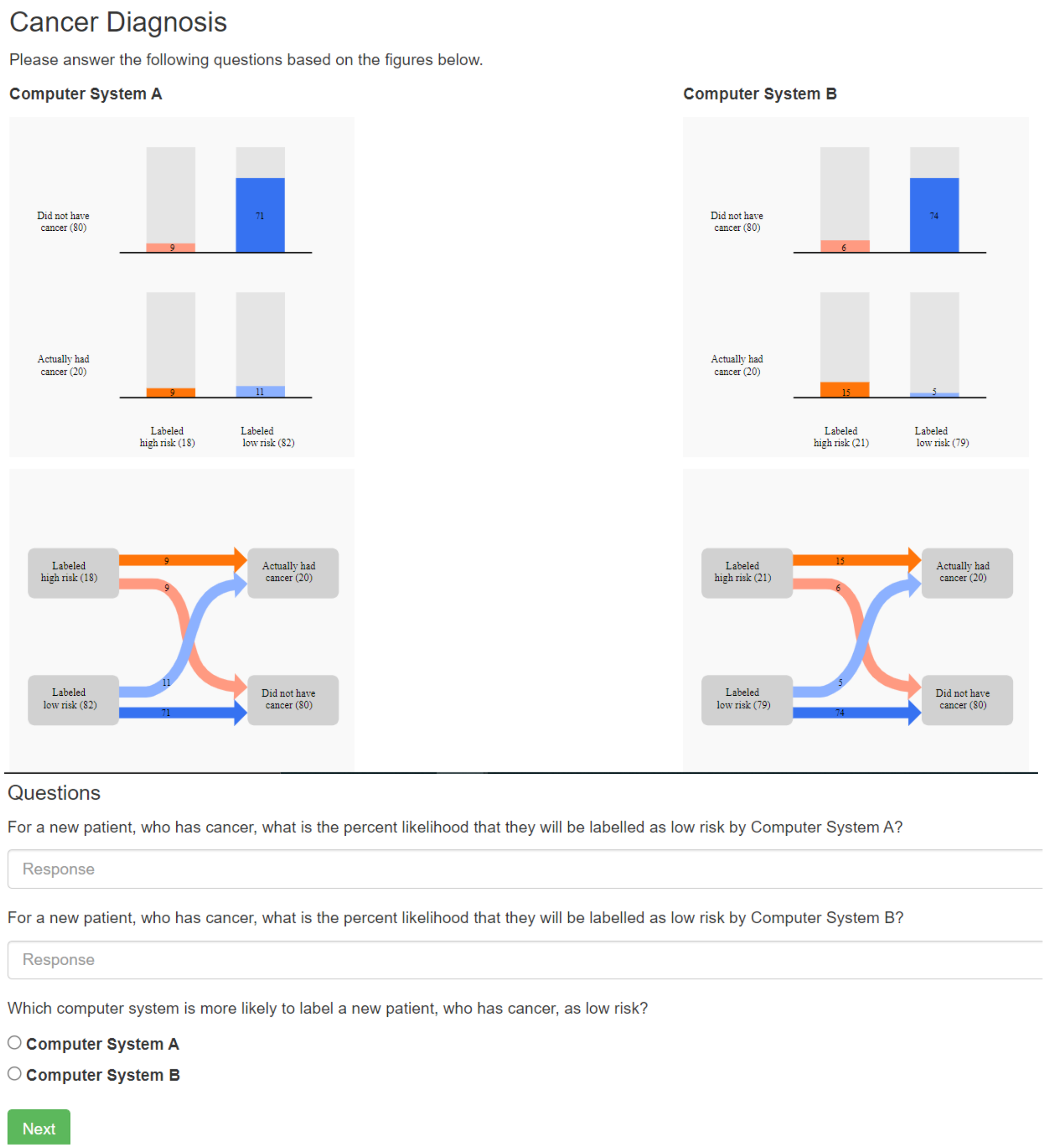
Response

In Computer System A, among patients who were labelled as high risk, what percentage did not have cancer?

Response

Next

Figure 8.7: Two confusion matrices side by side. This page asks questions about *comparing* the values in the two confusion matrices.



Cancer Diagnosis

Please answer the following questions based on the figures below.

Computer System A

Computer System B

Questions

For a new patient, who has cancer, what is the percent likelihood that they will be labelled as low risk by Computer System A?

Response

For a new patient, who has cancer, what is the percent likelihood that they will be labelled as low risk by Computer System B?

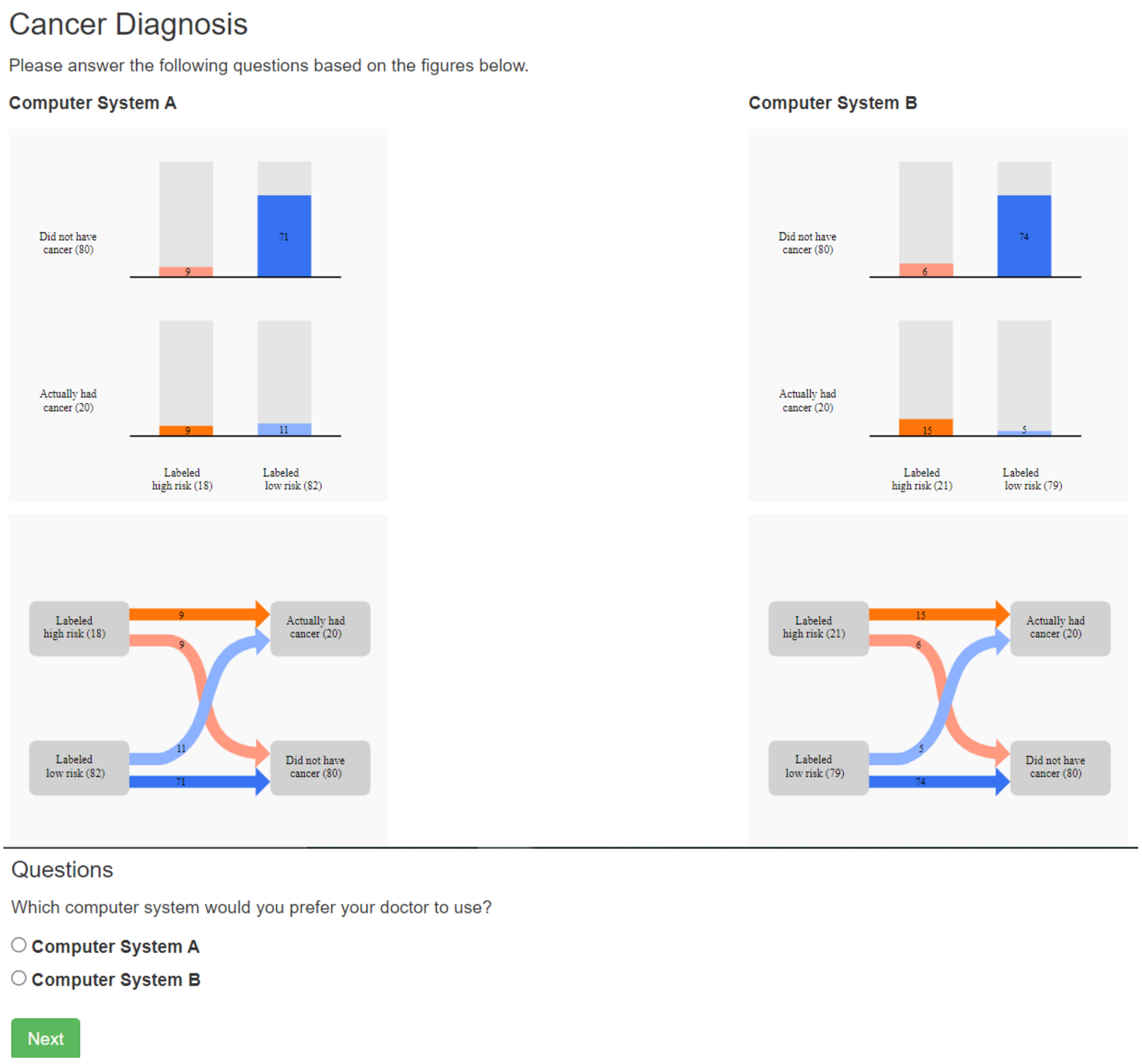
Response

Which computer system is more likely to label a new patient, who has cancer, as low risk?

○ **Computer System A**

○ **Computer System B**

Figure 8.8: Two confusion matrices side by side. This page asks questions about *simulating* a scenario based on the values in the two confusion matrices.



Cancer Diagnosis

Please answer the following questions based on the figures below.

Computer System A

Computer System B

Questions

Which computer system would you prefer your doctor to use?

○ **Computer System A**

○ **Computer System B**

Figure 8.9: Two hypothetically created confusion matrices side by side. This page asks questions about *pairwise comparison* of the confusion matices. Since one of the confusion matrices is worse in both false positives and false negatives, this question has a definitive answer.

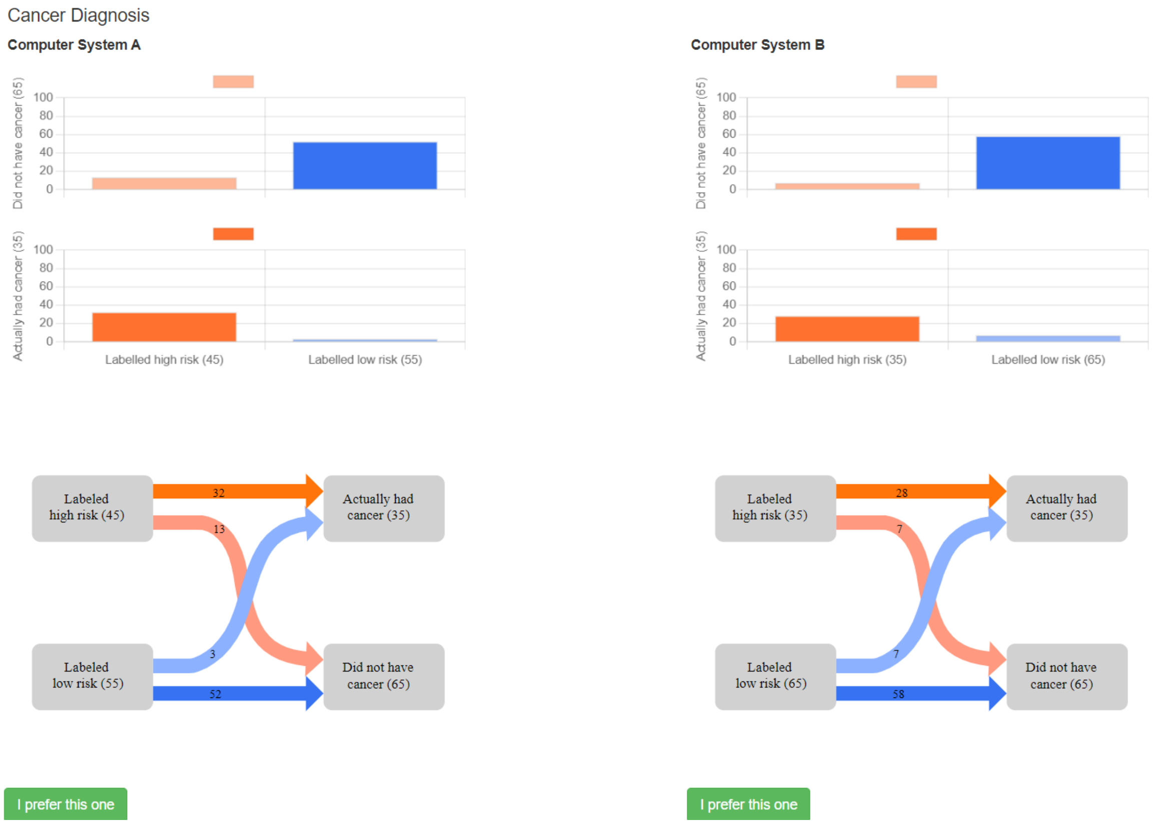


Figure 8.10: A sample of a pairwise comparison query from a run of the binary-search based procedure Algorithm 3.1.

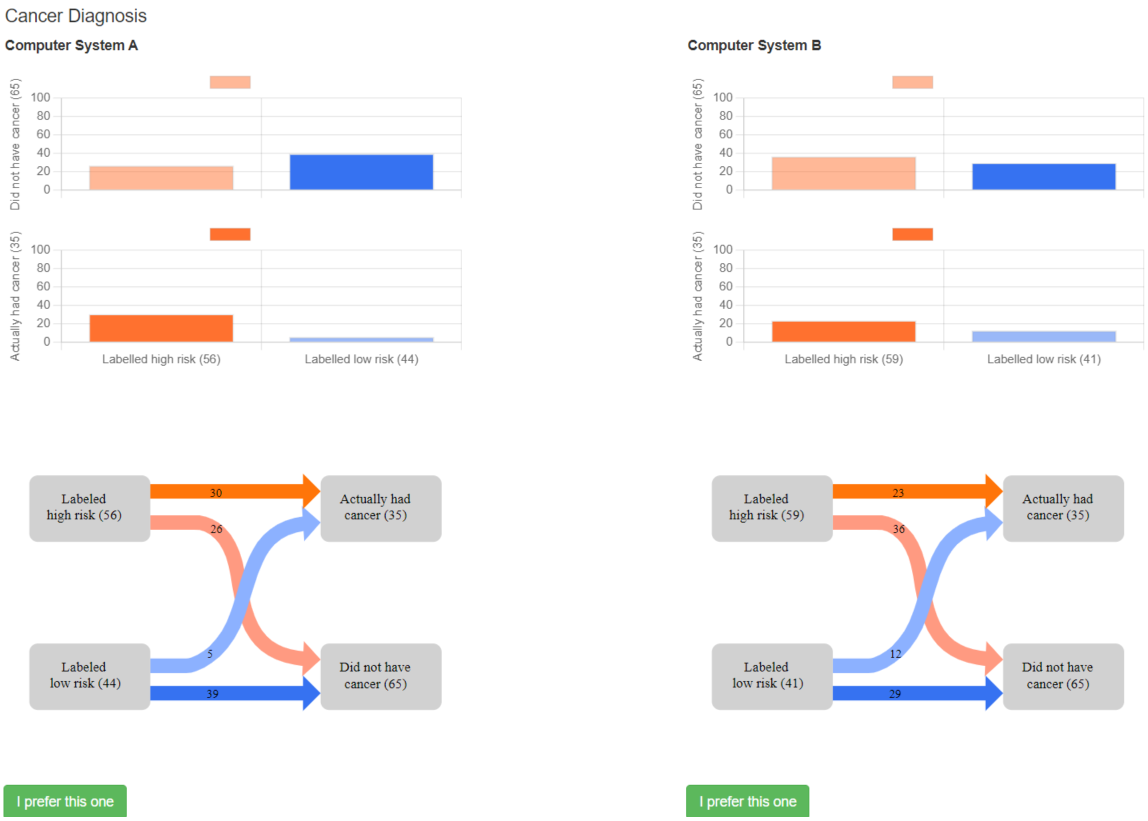


Figure 8.11: A sample of a pairwise comparison query comprising of randomly selected confusion matrices in the feasible region. These queries are used in evaluating the quality of the recovered metric.

CHAPTER 9: CONCLUSION AND FUTURE WORK

Typical default metrics in machine learning, such as accuracy applied to classification tasks, may not capture tradeoffs relevant to the problem at hand. Thus, optimizing such default metrics can have an undesirable impact on short and long-term utility, including the fairness of the resulting predictions across sensitive subgroups since the same issues plague default fairness measures. In this thesis, we formalized the problem of *Metric Elicitation (ME)* and proposed it as a principled framework for determining supervised classification metrics from user feedback. Through theoretical and empirical avenues, we showed that under certain conditions metric elicitation is equivalent to learning preferences between pairs of classifier statistics.

When the underlying metric is linear in the binary classification setup, we proposed an elicitation strategy to recover the oracle's metric, whose query complexity decays logarithmically with the desired resolution. We also showed that our query-complexity rates match the lower bound. We further extended our strategies to eliciting linear-fractional binary classification performance metrics.

We then broadened the scope of metric elicitation by proposing ME strategies for the more complicated multiclass classification setting. We proposed two algorithms for multiclass classification metric elicitation that use multiple binary-search subroutines that recover the oracle's linear metric. One of the proposed algorithms assumes that the oracle's metric is dependent on only the diagonal entries of the confusion matrices (a unique sparsity condition on the metric), and thus is useful when the number of classes is large. Similar to the binary case, we further provided algorithms for eliciting linear-fractional multiclass classification performance metrics.

With respect to applications to fairness, we devised a novel strategy to elicit group-fair performance metrics for multiclass classification problems with multiple sensitive groups that also includes selecting the trade-off between predictive performance and fairness violation. The procedure exploited the *piecewise* linearity of the metric in group-specific predictive rates, used binary-search based subroutines, and recovered the metric with linear query complexity. It was interesting to note that we were able to elicit a non-linear metric while maintaining the same query complexity order (linear in the number of unknowns) as the linear elicitation case.

We then used the tools and geometric characterizations build so far to solve three important problems that benefit the practical aspects of the proposed ME framework. The first involved increasing the complexity of the elicited metrics. The second was to exploit the current linear elicitation framework so to train deep neural networks for optimizing black-box metrics. The third was to conduct real-user study in order to elicit real-user metrics and reflect on the practical nuances of the ME framework. We draw out conclusions from each of these applications below.

The ME strategies for linear or quasi-linear functions of classifier statistics, can be restrictive in domains where the metrics are more complex and nuanced. Thus, we proposed novel strategies for eliciting metrics defined by *quadratic* functions of classifier statistics, which can easily be applied to fair metric elicitation setups as well. We were thus able to handle a more general family of metrics that can better capture a practitioner's innate preferences. We further generalized quadratic elicitation strategy to higher-order polynomial functions. All our metric elicitation procedures were shown to be robust to both finite sample and oracle feedback noise.

We then considered learning to optimize a classification metric defined by a black-box function of the confusion matrix. We proposed the Frank Wolfe with Elicited Gradient (FW-EG) method for optimizing black-box metrics given query access to the evaluation metric on a small validation set. Our framework included common distribution shift settings as special cases, and unlike prior distribution correction strategies, was able to handle general non-linear metrics. We showed how to model and estimate the example weights, but more importantly, we exploited the fact that the example weights can be seen as a gradient for the metric and estimated through metric elicitation procedure in the presence of a *machine* oracle. Experiments on various label noise, domain shift, and fair classification setups confirmed that our proposal compares favorably to the state-of-the-art baselines for each application. We briefly discussed how this procedure can be extended to optimize black-box metrics in the presence of a *human* oracle providing pairwise comparison feedback.

Lastly, we created a web UI for eliciting binary classification performance metrics that incorporates enhanced visualizations of confusion matrices for obtaining pairwise feedback. We then conducted a preliminary user-study in the binary classification setup in order to elicit real-users' performance metrics. In the process, we touched upon several practical aspects related to ME. In particular, we focused on checking workflow of the practical implementation, found support for the hypothesis that the implicit user preferences can be quantified using pairwise comparison queries, tested assumptions regarding the noise model, worked around with finite samples, elicited actual performance metrics in real-life scenarios, and evaluated the quality of the recovered metric. Using the quantitative and qualitative results from the pilot study, we formulated several guidelines and recommendations for practically implementing the metric elcitiation framework.

We envision the problem of *metric elicitation* to be an important, interesting, and chal-

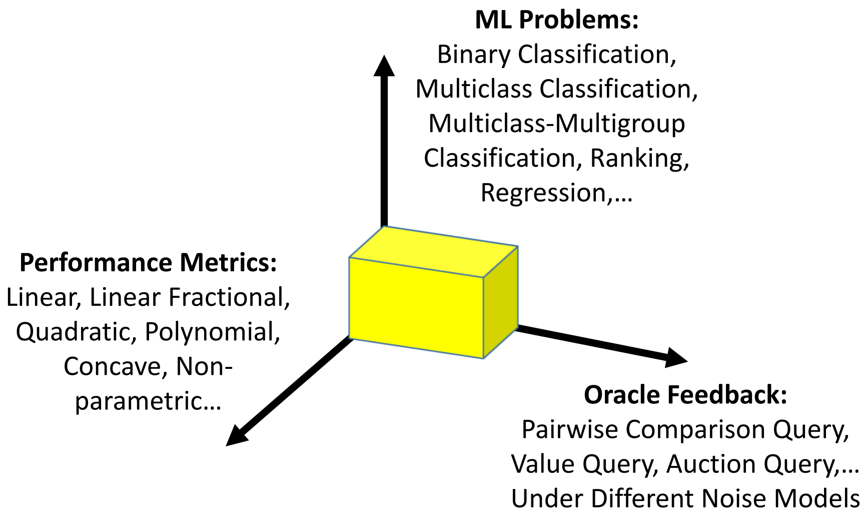


Figure 9.1: *Metric Elicitation for Predictive Machine Learning - Vision:* The three axes show three different nuances of metric elicitation. The first axis contain different predictive machine learning problems. On the second axis, there are various forms of performance metrics that can be elicited. Several oracle feedback and noise models lie on the third axis. This thesis provides solution to the box (shown in yellow color) covering a few parts of the larger problem of metric elicitation.

lenging topic for the future with many practical applications in the broad field of artificial intelligence. The underlying space of open problems can be broken into three separate axes. The axes are shown in Figure 9.1. On the first axis, there are different predictive machine learning problems such as classification, regression, ranking, etc. Each type of predictive problem involves new frontiers to be explored and exploited like we have done in this manuscript. For example, to elicit ranking metrics, one may require a thorough understanding of the space of statistics that summarize ranking effects. On the second axis, one may deal with various functional forms of performance metrics that can be elicited. Currently, we have focused on eliciting quasi-linear and polynomial functions of classifier statistics. Metric elicitation becomes much more challenging yet more practical when the functional forms are not assumed. The third axis stretches to different forms of oracle queries including various noise models. This direction guarantees the applicability of metric elicitation for real-world scenarios. The expected contribution in the future would be to solve the entire space of problems comprising the three axes, which may then result in a separate sub-field of artificial intelligence under the name – *Metric Elicitation for Predictive Machine Learning.* Once the metrics are elicited, sophisticated methods may be created to optimize those metrics similar to Chapter 7. Thus this entire line of work will answer important open questions in machine learning, impact several multi-disciplinary applications, and transform the way machine learning systems are deployed in practice.

APPENDIX A: BINARY CLASSIFICATION PERFORMANCE METRIC ELICITATION

A.1 VISUALIZING THE SET OF CONFUSION MATRICES

To clarify the geometry of the feasible set, we visualize one instance of the set of confusion matrices $\mathcal{C}$ using the dual representation of the supporting hyperplanes. The steps are:

1. *Population Model:* We assume a joint probability for $\mathcal{X} = [-1, 1]$ and $\mathcal{Y} = \{0, 1\}$ given by

$$f_X = \mathbb{U}[-1, 1] \quad \text{and} \quad \eta(x) = \frac{1}{1 + e^{ax}}, \tag{A.1}$$

where $\mathbb{U}[-1, 1]$ is the uniform distribution on $[-1, 1]$ and $a > 0$ is a parameter controlling the degree of noise in the labels. If a is large, then with high probability, the true label is 1 on [-1, 0] and 0 on [0, 1]. On the contrary, if a is small, then there are no separable regions and the classes are mixed in $[-1, 1]$.

Furthermore, the integral $\int_{-1}^{1} \frac{1}{1+e^{ax}} dx = 1$ for $a \in \mathbb{R}$ implying $\mathbb{P}(Y = 1) = \zeta = \frac{1}{2} \; \forall \; a \in \mathbb{R}$.

2. *Generate Hyperplanes:* Take $\theta \in [0, 2\pi]$ and set $\mathbf{m} = (m_{11}, m_{00}) = (\cos\theta, \sin\theta)$. Let us denote x' as the point where the probability of positive class $\eta(x)$ is equal to the optimal threshold of Proposition 3.1. Solving for x in the equation $1/(1 + e^{ax}) = m_{00}/(m_{00} + m_{11})$ gives us

$$x' = \Pi_{[-1,1]}\left\{\tfrac{1}{a} \ln\left(\tfrac{m_{11}}{m_{00}}\right)\right\}, \tag{A.2}$$

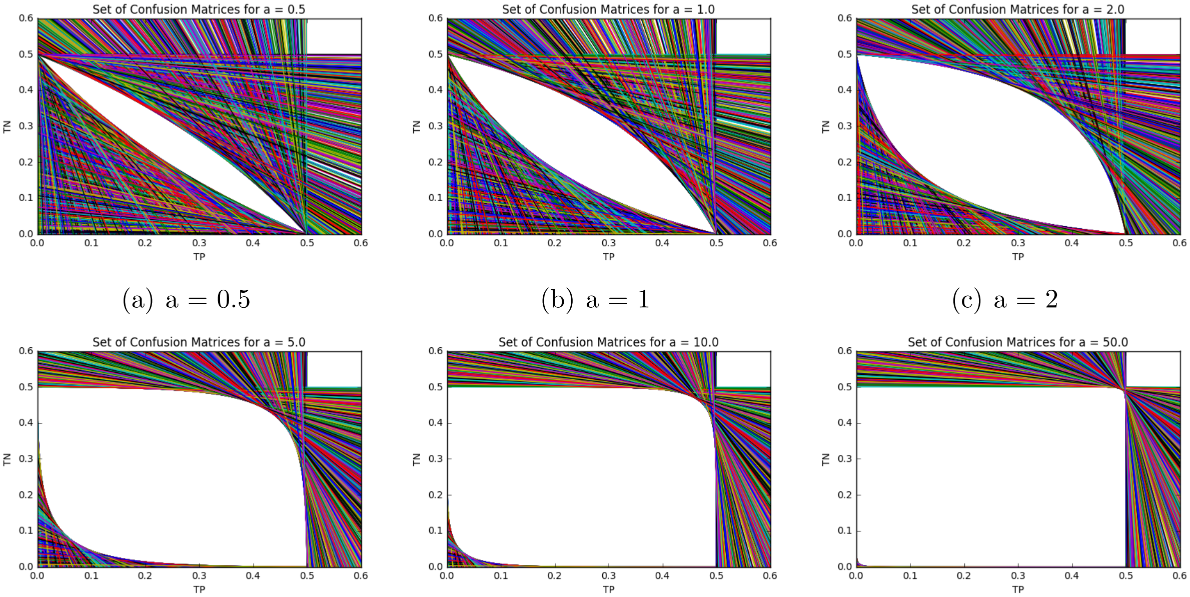
where $\Pi_{[-1,1]}\{z\}$ is the projection of z on the interval $[-1, 1]$. If $m_{11} + m_{00} \geq 0$, then the Bayes classifier $\bar{h}$ predicts class 1 on the region $[-1, x']$ and 0 on the remaining region. If $m_{11} + m_{00} < 0$, $\bar{h}$ does the opposite. Using the fact that $Y|X$ and $\bar{h}|X$ are independent, we have that

(a) if $m_{11} + m_{00} \geq 0$, then

$$\overline{TP}_{\mathbf{m}} = \frac{1}{2} \int\limits_{-1}^{x'} \tfrac{1}{1+e^{ax}} dx, \qquad \overline{TN}_{\mathbf{m}} = \tfrac{1}{2} \int\limits_{x'}^{1} \tfrac{e^{ax}}{1+e^{ax}} dx. \tag{A.3}$$

(b) if $m_{11} + m_{00} < 0$, then

$$\overline{TP}_{\mathbf{m}} = \frac{1}{2} \int\limits_{x'}^{1} \tfrac{1}{1+e^{ax}} dx, \qquad \overline{TN}_{\mathbf{m}} = \tfrac{1}{2} \int\limits_{-1}^{x'} \tfrac{e^{ax}}{1+e^{ax}} dx. \tag{A.4}$$



(a) a = 0.5

(b) a = 1

(c) a = 2

(d) a = 5

(e) a = 10

(f) a = 50

Figure A.1: Supporting hyperplanes and associated set of feasible confusion matrices for exponential model described in equation (A.1) with $a = 0.5, 1, 2, 5, 10$ and 50. The middle white region is $\mathcal{C}$, which is the intersection of half-spaces associated with its supporting hyperplanes.

Now, we can obtain the hyperplane as defined in (3.19) for each θ. We sample around thousand $\theta's \in [0, 2\pi]$ randomly. We then obtain the hyperplanes following the above process and plot them.

The sets of feasible confusion matrices $\mathcal{C}$'s for $a = 0.5, 1, 2, 5, 10$, and 50 are shown in Figure A.1. The middle white region is $\mathcal{C}$: the intersection of the half-spaces associated with its supporting hyperplanes. The curve on the right corresponds to the confusion matrices on the upper boundary $\partial\mathcal{C}_+$. Similarly, the curve on the left corresponds to the confusion matrices on the lower boundary $\partial\mathcal{C}_-$. Points $(\zeta, 0) = (\frac{1}{2}, 0)$ and $(0, 1-\zeta) = (0, \frac{1}{2})$ are the two vertices. The geometry is 180-degree rotationally symmetric around the center point $(\frac{1}{4}, \frac{1}{4})$, which corresponds to the confusion matrix of the uniform random classifier, i.e., the classifier which predicts both classes with equal probability for any input.

Notice that as we increase the separability of the two classes via a, all the points in $[0, \zeta] \times [0, 1-\zeta]$ becomes feasible. In other words, if the data is completely separable, then the corners on the top-right and the bottom left are achievable. If the data is 'inseparable', then the feasible set contains only the diagonal line joining $(0, \frac{1}{2})$ and $(\frac{1}{2}, 0)$, which passes through $(\frac{1}{4}, \frac{1}{4})$.

A.2 PROOFS

Lemma A.1. The feasible set of confusion matrices $\mathcal{C}$ has the following properties:

(i). For all $(TP, TN) \in \mathcal{C}$, $0 \leq TP \leq \zeta$, and $0 \leq TN \leq 1 - \zeta$.

(ii). $(\zeta, 0) \in \mathcal{C}$ and $(0, 1 - \zeta) \in \mathcal{C}$.

(iii). For all $(TP, TN) \in \mathcal{C}$, $(\zeta - TP, 1 - \zeta - TN) \in \mathcal{C}$.

(iv). $\mathcal{C}$ is convex.

(v). $\mathcal{C}$ has a supporting hyperplane associated to every normal vector.

(vi). Any supporting hyperplane with positive slope is tangent to $\mathcal{C}$ at $(0, 1 - \zeta)$ or $(\zeta, 0)$.

Proof. We prove the statements as follows:

(i). $0 \leq \mathbb{P}[h = Y = 1] \leq \mathbb{P}[Y = 1] = \zeta$, and similarly, $0 \leq \mathbb{P}[h = Y = 0] \leq \mathbb{P}[Y = 0] = 1 - \zeta$.

(ii). If h is the trivial classifier which always predicts 1, then $TP(h) = \Pr[h = Y = 1] = \Pr[Y = 1] = \zeta$, and $TN(h) = 0$. This means that $(\zeta, 0) \in \mathcal{C}$. Similarly, if h is the classifier which always predicts 0, then $TP(h) = \Pr[h = Y = 1] = 0$, and $TN(h) = \Pr[h = Y = 0] = \Pr[Y = 0] = 1 - \zeta$. Therefore, $(0, 1 - \zeta) \in \mathcal{C}$.

(iii). Let h be a classifier such that $TP(h) = TP$, $TN(h) = TN$. Now, consider the classifier $1 - h$ (which predicts exactly the opposite of h). We have that

$$\begin{aligned} TP(1-h) &= \mathbb{P}[(1-h) = Y = 1] \\ &= \mathbb{P}[Y = 1] - \mathbb{P}[h = Y = 1] \\ &= \zeta - TP(h). \end{aligned} \tag{A.5}$$

A similar argument gives

$$TN(1-h) = 1 - \zeta - TN(h). \tag{A.6}$$

(iv). Consider any two confusion matrices (TP_1, TN_1), $(TP_2, TN_2) \in \mathcal{C}$, attained by the classifiers $h_1, h_2 \in \mathcal{H}$, respectively. Let $0 \leq \lambda \leq 1$. Define a classifier h' which predicts the output from the classifier h_1 with probability λ and predicts the output of the classifier h_2 with probability $1 - \lambda$. Then,

$$
\begin{aligned}
TP(h') &= \mathbb{P}[h' = Y = 1] \\
&= \mathbb{P}[h_1 = Y = 1|h = h_1]\mathbb{P}[h = h_1] + \mathbb{P}[h_2 = Y = 1|h = h_2]\mathbb{P}[h = h_2] \quad (A.7) \\
&= \lambda TP(h_1) + (1-\lambda)TP(h_2). \quad (A.8)
\end{aligned}
$$

A similar argument gives the convex combination for TN. Thus, $\lambda(TP(h_1), TN(h_1)) + (1-\lambda)(TP(h_2), TN(h_2)) \in \mathcal{C}$ and hence, $\mathcal{C}$ is convex.

(v). This follows from convexity (iv) and boundedness (i).

(vi). For any bounded, convex region in $[0,\zeta]\times[0,1-\zeta]$ which contains the points $(0,\zeta)$ and $(0,1-\zeta)$, it is true that any positively sloped supporting hyperplane will be tangent to $(0,\zeta)$ or $(0,1-\zeta)$.

QED.

Lemma A.2. The boundary of $\mathcal{C}$ is exactly the confusion matrices of estimators of the form $\lambda\mathbf{1}[\eta(x) \geq t] + (1-\lambda)\mathbf{1}[\eta(x) > t]$ and $\lambda\mathbf{1}[\eta(x) < t] + (1-\lambda)\mathbf{1}[\eta(x) \leq t]$ for some $\lambda, t \in [0,1]$.

Proof. To prove that the boundary is attained by estimators of these forms, consider solving the problem under the constraint $\mathbb{P}[h=1] = c$. We have $\mathbb{P}[h=1] = TP + FP$, and $\zeta = \mathbb{P}[Y=1] = TP + FN$, so we get

$$
TP - TN \;=\; c + \zeta - TP - TN - FP - FN \;=\; c + \zeta - 1, \tag{A.9}
$$

which is a constant. Note that no confusion matrix has two values of $TP - TN$. This effectively partitions $\mathcal{C}$, since all confusion matrices are attained by varying c from 0 to 1. Furthermore, since $A := TN = TP - c - \zeta + 1$ is an affine space (a line in tp-tn coordinate system), $\mathcal{C} \cap A$ has at least one endpoint, because A would pass through the box $[\zeta, 0] \times [0, 1-\zeta]$ and has at most two endpoints due to convexity and boundedness of $\mathcal{C}$. Since A is a line with positive slope, $\mathcal{C} \cap A$ is a single point only when A is tangent to $\mathcal{C}$ at $(0, 1-\zeta)$ or $(\zeta, 0)$, from Lemma A.1, part (vi).

Since the affine space A has positive slope, we claim that the two endpoints are attained by maximizing or minimizing $TP(h)$ subject to $\Pr[h = 1] = c$. It remains to show that this happens for estimators of the form $h^{\lambda}_{t+} := \lambda\mathbf{1}[\eta(x) \geq t] + (1-\lambda)\mathbf{1}[\eta(x) > t]$ and $h^{\lambda}_{t-} := \lambda\mathbf{1}[\eta(x) < t] + (1-\lambda)\mathbf{1}[\eta(x) \leq t]$, respectively.

Let h be any estimator, and recall

$$
TP(h) := \int_{\mathcal{X}} \eta(x)\mathbb{P}[h = 1|X = x]\,\mathrm{d}f_X. \tag{A.10}
$$

It should be clear that under a constraint $\mathbb{P}[h = 1] = c$, the optimal choice of h puts all the weight onto the larger values of η. One can begin by classifying those X into the positive class where $n(X)$ is maximum, until one exhausts the budget of c. Let t be such that $\mathbb{P}[h^0_{t+} = 1] \leq c \leq \mathbb{P}[h^1_{t+} = 1]$, and let $\lambda \in [0, 1]$ be chosen such that $\mathbb{P}[h^\lambda_{t+} = 1] = c$, then h^λ_{t+} must maximize $TP(h)$ subject to $\mathbb{P}[h = 1] = c$.

A similar argument shows that all TP-minimizing boundary points are attained by the h_{t-}'s. QED.

Remark A.1. Under Assumption 3.1, $\mathbf{1}[\eta(x) > t] = \mathbf{1}[\eta(x) \geq t]$ and $\mathbf{1}[\eta(x) < t] = \mathbf{1}[\eta(x) \leq t]$. Thus, the boundary of $\mathcal{C}$ is the confusion matrices of estimators of the form $\mathbf{1}[\eta(x) \geq t]$ and $\mathbf{1}[\eta(x) \leq t]$ for some $t \in [0, 1]$.

Proof of Proposition 3.1. Note, we are maximizing a linear function on a convex set. There are 6 cases to consider:

1. If the signs of m_{11} and m_{00} differ, the maximum is attained either at $(0, 1 - \zeta)$ or $(\zeta, 0)$, as per Lemma A.1, part (vi). Which of the two is optimum depends on whether $|m_{11}| \geq |m_{00}|$, i.e. on the sign of $m_{11} + m_{00}$. It should be easy to check that in all four possible cases, the statement holds, noting that in all four cases, $0 \leq m_{00}/(m_{11} + m_{00}) \leq 1$.

2. If $m_{11}, m_{00} \geq 0$, then the maximum is attained on $\partial\mathcal{C}_+$, and the proof below gives the desired result.

   We know, from Lemma A.2, that $\bar{h}$ must be of the form $\mathbf{1}[\eta(x) \geq t]$ for some t. It suffices to find t. Thus, we wish to maximize $m_{11}TP(h_t) + m_{00}TN(h_t)$. Now, let $Z := \eta(X)$ be the random variable obtained by evaluating η at random X. Under Assumption 3.1, $df_X = df_Z$ and we have that

   $$TP(h_t) \;= \int_{x:\eta(x)\geq t} \eta(x)\,\mathrm{d}f_X \;= \int_t^1 z\,\mathrm{d}f_Z. \tag{A.11}$$

   Similarly, $TN(h_t) = \int_0^t (1 - z)\,\mathrm{d}f_Z$. Therefore,

   $$\tfrac{\partial}{\partial t}\big(m_{11}TP(h_t) + m_{00}TN(h_t)\big) = -m_{11}tf_Z(t) + \cdot m_{00}(1 - t)f_Z(t). \tag{A.12}$$

   So, the critical point is attained at $t = m_{00}/(m_{11} + m_{00})$, as desired. A similar argument gives the converse result for $m_{11} + m_{00} < 0$.

3. if $m_{11}, m_{00} < 0$, then the maximum is attained on $\partial\mathcal{C}_-$, and an argument identical to the proof above gives the desired result.

QED.

Proof of Proposition 3.2. That $\mathcal{C}$ is convex and bounded is already proven in Lemma A.1. To see that $\mathcal{C}$ is closed, note that, from Lemma A.2, every boundary point is attained. From Lemma A.1, part (iii), it follows that $\mathcal{C}$ is 180-degree rotationally symmetric around the point $(\frac{\zeta}{2}, \frac{1-\zeta}{2})$.

Further, recall every boundary point of $\mathcal{C}$ can be attained by a thresholding estimator. By the discussion in Section 3.2, every boundary point is the optimal classifier for some linear performance metric, and the vector defining this linear metric is exactly the normal vector of the supporting hyperplane at the boundary point.

A vertex exists if (and only if) some point is supported by more than one tangent hyperplane in two dimensional space. This means it is optimal for more than one linear metric. Clearly, all the hyperplanes corresponding to the slope of the metrics where m_{11} and m_{00} are of opposite sign (i.e. hyperplanes with positive slope) support either $(\zeta, 0)$ or $(0, 1-\zeta)$. So, there are at least two supporting hyperplanes at these points, which make them the vertices. Now, it remains to show that there are no other vertices for the set $\mathcal{C}$.

Now consider the case when the slopes of the hyperplanes are negative, i.e. m_{11} and m_{00} have the same sign for the corresponding linear metrics. We know from Proposition 3.1 that optimal classifiers for linear metrics are threshold classifiers. Therefore there exist more than one threshold classifier of the form $h_t = \mathbf{1}[\eta(x) \geq t]$ with the same confusion matrix. Let's call them h_{t_1} and h_{t_2} for the two thresholds $t_1, t_2 \in [0, 1]$. This means that

$$\int_{x:\eta(x)\geq t_1} \eta(x) df_X = \int_{x:\eta(x)\geq t_2} \eta(x) df_X. \tag{A.13}$$

Hence, there are multiple values of η which are never attained! This contradicts that g is strictly decreasing. Therefore, there are no vertices other than $(\zeta, 0)$ or $(0, 1-\zeta)$ in $\mathcal{C}$.

Now, we show that no supporting hyperplane is tangent at multiple points (i.e., there no flat regions on the boundary). If suppose there is a hyperplane which supports two points on the boundary. Then there exist two threshold classifiers with arbitrarily close threshold values, but confusion matrices that are well-separated. Therefore, there must exist some value of η which exists with non-zero probability, contradicting the continuity of g. By the discussion above, we conclude that under Assumption 3.1, every supporting hyperplane to the convext set $\mathcal{C}$ is tangent to only one point. This makes the set $\mathcal{C}$ strictly convex. QED.

Proof of Lemma 3.1. We will prove the result for $\phi \circ \rho^+$ on $\partial\mathcal{C}^+$, and the argument for $\psi \circ \rho^-$ on $\partial\mathcal{C}^+$ is essentially the same. For simplicity, we drop the + symbols in the notation. Recall that a function is quasiconcave if and only if its superlevel sets are convex.

It is given that ϕ is quasiconcave. Let S be some superlevel set of ϕ. We first want to show that for any $r < s < t$, if $\rho(r) \in S$ and $\rho(t) \in S$, then $\rho(s) \in S$. Since ρ is a continuous bijection, due to the geometry of $\mathcal{C}$ (Lemma A.1 and Proposition 3.2), we must have — without loss of generality — $TP(\rho(r)) < TP(\rho(s)) < TP(\rho(t))$, and $TN(\rho(r)) > TN(\rho(s)) > TN(\rho(t))$. (otherwise swap r and t). Since the set $\mathcal{C}$ is strictly convex and the image of ρ is $\partial\mathcal{C}$, then $\rho(s)$ must dominate (component-wise) a point in the convex combination of $\rho(r)$ and $\rho(t)$. Say that point is z. Since ϕ is monotone increasing, then $x \in S \implies y \in S$ for all $y \geq x$ componentwise. Thereofore, $\phi(\rho(s)) \geq \phi(z)$. Since, S is convex, $z \in S$ and, due to the argument above, $\rho(s) \in S$.

This implies that $\rho^{-1}(\partial\mathcal{C} \cap S)$ is an interval, and is therefore convex. Thus, the superlevel sets of $\phi \circ \rho$ are convex, so it is quasiconcave, as desired. This implies unimodaltiy as a function over the real line which has more than one local maximum can not be quasiconcave (consider the super-level set for some value slightly less than the lowest of the two peaks). QED.

Proof of Proposition 3.3. For this proof, we denote TP and TN as C_{11} and C_{00}, respectively. Let us take a linear-fractional metric

$$\phi(C) = \frac{p_{11}C_{11} + p_{00}C_{00} + p_0}{q_{11}C_{11} + q_{00}C_{00} + q_0} \tag{A.14}$$

where $p_{11}, q_{11}, p_{00}, q_{00}$ are not zero simultaneously. We want $\phi(C)$ to be monotonic in TP, TN and bounded. If for any $C \in \mathcal{C}$, $\phi(C) < 0$, we can add a large positive constant such that $\phi(C) \geq 0$, and still the metric would remain linear fractional. So, it is sufficient to assume $\phi(C) \geq 0$. Furthermore, boundedness of ϕ implies $\phi(C) \in [0, D]$, for some $\mathbb{R} \ni D \geq 0$. Therefore, we may divide $\phi(C)$ by D so that $\phi(C) \in [0, 1]$ for all $C \in \mathcal{C}$. Still, the metric is linear fractional and $\phi(C) \in [0, 1]$.

Taking derivative of $\phi(C)$ w.r.t. C_{11}.

$$\frac{\partial\phi(C)}{\partial C_{11}} = \frac{p_{11}}{q_{11}C_{11} + q_{00}C_{00} + q_0} - \frac{q_{11}(p_{11}C_{11} + p_{00}C_{00} + p_0)}{(q_{11}C_{11} + q_{00}C_{00} + q_0)^2} \geq 0 \tag{A.15}$$

$$\Rightarrow p_{11}(q_{11}C_{11} + q_{00}C_{00} + q_0) \geq q_{11}(p_{11}C_{11} + p_{00}C_{00} + p_0) \tag{A.16}$$

If denominator is positive then the numerator is positive as well.

- Case 1: The denominator $q_{11}C_{11} + q_{00}C_{00} + q_0 \geq 0$.
  - Case (a) $q_{11} > 0$.

$$\begin{aligned}&\Rightarrow p_{11} \geq q_{11}\phi(C)\\&\Rightarrow p_{11} \geq q_{11}\sup_{C\in\mathcal{C}}\phi(C)\\&\Rightarrow p_{11} \geq q_{11}\overline{\tau} \qquad \text{(Necessary Condition)}\end{aligned} \tag{A.17}$$

We are considering sufficient condition, which means $\overline{\tau}$ can vary from $[0,1]$. Hence, a sufficient condition for monotonicity in C_{11} is $p_{11} \geq q_{11}$. Furthermore, $p_{11} \geq 0$ as well.

- Case (b) $q_{11} < 0$.

$$\Rightarrow p_{11} \geq q_{11}\overline{\tau} \tag{A.18}$$

Since $q_{11} < 0$ and $\overline{\tau} \in [0,1]$, sufficient condition is $p_{11} \geq 0$. So, in this case as well we have that

$$p_{11} \geq q_{11},\ p_{11} \geq 0. \tag{A.19}$$

- Case(c) $q_{11} = 0$.

$$\Rightarrow p_{11} \geq 0 \tag{A.20}$$

We again have $p_{11} \geq q_{11}$ and $p_{11} \geq 0$ as sufficient conditions.

A similar case holds for C_{00}, implying $p_{00} \geq q_{00}$ and $p_{00} \geq 0$.

- Case 2: The denominator $q_{11}C_{11} + q_{00}C_{00} + q_0$ is negative.

$$\begin{aligned}p_{11} &\leq q_{11}\Big(\frac{p_{11}C_{11} + p_{00}C_{00} + p_0}{q_{11}C_{11} + q_{00}C_{00} + q_0}\Big)\\ \Rightarrow p_{11} &\leq q_{11}\overline{\tau}\end{aligned} \tag{A.21}$$

- Case(a) If $q_{11} > 0$. So, we have $p_{11} \leq q_{11}$ and $p_{11} \leq 0$ as sufficient condition.
- Case(b) If $q_{11} < 0$, $\Rightarrow p_{11} \leq q_{11}$. So, we have $q_{11} < 0$, $\Rightarrow p_{11} < 0$ as sufficient condition.
- Case(c) If $q_{11} = 0$, $\Rightarrow p_{11} \leq 0$ and $p_{11} \leq q_{11}$ as sufficient condition.

So in all the cases we have that

$$p_{11} \leq q_{11} \text{ and } p_{11} \leq 0 \tag{A.22}$$

as the sufficient conditions. A similar case holds for C_{00} resulting in $p_{00} \leq q_{00}$ and $p_{00} \leq 0$.

Suppose the points where denominator is positive is $\mathcal{C}^{+} \subseteq \mathcal{C}$. Suppose the points where denominator is negative is $\mathcal{C}^{-} \subseteq \mathcal{C}$. For gradient to be non-negative at points belonging to $\mathcal{C}^{+}$, the sufficient condition is

$$\begin{aligned} &p_{11} \geq q_{11} \text{ and } p_{11} \geq 0 \\ &p_{00} \geq q_{00} \text{ and } p_{00} \geq 0 \end{aligned} \tag{A.23}$$

For gradient to be non-negative at points belonging to $\mathcal{C}^{-}$, the sufficient condition is

$$\begin{aligned} &p_{11} \leq q_{11} \text{ and } p_{11} \leq 0 \\ &p_{00} \leq q_{00} \text{ and } p_{00} \leq 0 \end{aligned} \tag{A.24}$$

If $\mathcal{C}_{+}$ and $\mathcal{C}_{-}$ are not empty sets, then the gradient is non-negative only when $p_{11}, p_{00} = 0$ and $q_{11}, q_{00} = 0$. This is not possible by the definition described in (A.14). Hence, one of $\mathcal{C}_{+}$ or $\mathcal{C}_{-}$ should be empty. WLOG, we assume $\mathcal{C}_{-}$ is empty and conclude that $\mathcal{C}_{+} = \mathcal{C}$.
An immediate consequence of this is, WLOG, we can take both the numerator and the denominator to be positive, and the sufficient conditions for monotonicity are as follows:

$$\begin{aligned} &p_{11} \geq q_{11} \text{ and } p_{11} \geq 0 \\ &p_{00} \geq q_{00} \text{ and } p_{00} \geq 0 \end{aligned} \tag{A.25}$$

Now, let us take a point in the feasible space $(\zeta, 0)$. We know that

$$\begin{aligned} \phi((\zeta,0)) = \frac{p_{11}\zeta + p_0}{q_{11}\zeta + q_0} &\leq \bar{\tau} \\ &\Rightarrow p_{11}\zeta + p_0 \leq \bar{\tau}(q_{11}\zeta + q_0) \\ &\Rightarrow (p_{11} - \bar{\tau}q_{11})\zeta + (p_0 - \bar{\tau}q_0) \leq 0 \\ &\Rightarrow (p_0 - \bar{\tau}q_0) \leq -\underbrace{(p_{11} - \bar{\tau}q_{11})}_{\text{positive}}\underbrace{\zeta}_{\text{positive}} \\ &\Rightarrow (p_0 - \bar{\tau}q_0) \leq 0. \end{aligned} \tag{A.26}$$

Metric being bounded in $[0,1]$ gives us

$$\begin{aligned} &\frac{p_{11}C_{11} + p_{00}C_{00} + p_0}{q_{11}C_{11} + q_{00}C_{00} + q_0} \leq 1 \\ \Rightarrow\ & p_{11}C_{11} + p_{00}C_{00} + p_0 \leq q_{11}C_{11} + q_{00}C_{00} + q_0 \end{aligned} \tag{A.27}$$

$$\Rightarrow q_0 \geq (p_{11} - q_{11})c_{11} + (p_{00} - q_{00})c_{00} + p_0 \qquad \forall C \in \mathcal{C}. \tag{A.28}$$

Hence, a sufficient condition is

$$q_0 = (p_{11} - q_{11})\zeta + (p_{00} - q_{00})(1 - \zeta) + p_0. \tag{A.29}$$

Equation (A.26), which we derived from monotonicity, implies that

- Case (a) $q_0 \geq 0, \Rightarrow p_0 \leq 0$ as a sufficient condition.
- Case (b) $q_0 \leq 0, \Rightarrow p_0 \leq q_0 \leq 0$ as a sufficient condition.

Since the numerator is positive for all $C \in \mathcal{C}$ and $p_{11}, p_{00} \geq 0$, a sufficient condition for p_0 is $p_0 = 0$.

Finally, a monotonic, bounded in $[0, 1]$, linear fractional metric is defined by

$$\phi(C) = \frac{p_{11}c_{11} + p_{00}c_{00} + p_0}{q_{11}c_{11} + q_{00}c_{00} + q_0}, \tag{A.30}$$

where $p_{11} \geq q_{11}, p_{11} \geq 0, p_{00} \geq q_{00}, p_{00} \geq 0, q_0 = (p_{11} - q_{11})\zeta + (p_{00} - q_{00})(1 - \zeta) + p_0, p_0 = 0$, and p_{11}, q_{11}, p_{00}, and q_{00} are not simulataneously zero. Further, we can divide the numerator and denominator with $p_{11} + p_{00}$ without changing the metric ϕ and the above sufficient conditions. Therefore, for elicitation purposes, we can take $p_{11} + p_{00} = 1$. QED.

Proof of Proposition 3.4. For this proof as well, we use $TP = C_{11}$ and $TN = C_{00}$. Since the linear fractional matrix is monotonically increasing in C_{11} and C_{00}, it is maximized at the upper boundary $\partial\mathcal{C}_+$. Hence $m_{11} \geq 0$ and $m_{00} \geq 0$. So, after running Algorithm 3.1, we get a hyperplane such that

$$\begin{aligned} p_{11} - \tau q_{11} = \alpha m_{11}, \quad p_{00} - \tau q_{00} = \alpha m_{00}, \\ p_0 - \tau q_0 = -\alpha \underbrace{(m_{11}C^*_{11} + m_{00}C^*_{00})}_{=:C_0}. \end{aligned} \tag{A.31}$$

Since $p_{11} - \bar{\tau} q_{11} \geq 0$ and $m_{11} \geq 0, \Rightarrow \alpha \geq 0$. As discussed in the main paper, we avoid the case when $\alpha = 0$. Therefore, we have that $\alpha > 0$.

Equation (A.31) implies that

$$\begin{aligned} \frac{p_{11}}{\alpha} - \frac{\tau q_{11}}{\alpha} = m_{11}, \quad \frac{p_{00}}{\alpha} - \frac{\tau q_{00}}{\alpha} = m_{00}, \\ \frac{p_0}{\alpha} - \frac{\tau q_0}{\alpha} = -C_0. \end{aligned} \tag{A.32}$$

Assume $p'_{11} = \frac{p_{11}}{\alpha}, p'_{00} = \frac{p_{00}}{\alpha}$, $q'_{11} = \frac{q_{11}}{\alpha}$, $q'_{00} = \frac{q_{00}}{\alpha}$, $p'_0 = \frac{p_0}{\alpha}$, $q'_0 = \frac{q_0}{\alpha}$. Then, the above system of equations turns into

$$\begin{aligned} p'_{11} - \overline{\tau} q'_{11} = m_{11}, \quad p'_{00} - \overline{\tau} q'_{00} = m_{00}, \\ p'_0 - \overline{\tau} q'_0 = -C_0. \end{aligned} \tag{A.33}$$

A ϕ' metric defined by the $p'_{11}, p'_{00}, q'_{11}, q'_{00}, q'_0$ is monotonic, bounded in $[0, 1]$, and satisfies all the sufficient conditions of Assumptions 3.2, i.e.,

$$\begin{aligned} p'_{11} \geq q'_{11} \ , \ p'_{00} \geq q'_{11}, \ p'_{11} \geq 0 \ , \ p'_{00} \geq 0, \\ q'_0 = (p'_{11} - q_{11})\pi + (p'_{00} - q'_{00})\pi + p'_0, \ p'_0 = 0. \end{aligned} \tag{A.34}$$

As discussed in Chapter 3, solving the above system does not harm the elicitation task. For simplicity, replacing the “ $'$ ” notation with the normal one, we have that

$$\begin{aligned} p_{11} - \overline{\tau} q_{11} = m_{11}, \quad p_{00} - \overline{\tau} q_{00} = m_{00}, \\ p_0 - \overline{\tau} q_0 = -C_0 \end{aligned} \tag{A.35}$$

From last equation, we have that $\overline{\tau} = \frac{C_0 + p_0}{q_0}$. Putting it in the rest gives us

$$q_0 p_{11} - (C_0 + p_0) q_{11} = m_{11} q_0 \quad \text{and} \quad q_0 p_{00} - (C_0 + p_0) q_{00} = m_{00} q_0. \tag{A.36}$$

We already have

$$\begin{aligned} q_0 &= (p_{11} - q_{11})\zeta + (p_{00} - q_{00})(1 - \zeta) + p_0 \\ \Rightarrow q_{11} &= \frac{p_{00}(1 - \zeta) - q_{00}(1 - \zeta) + p_{11}\zeta - q_0 + p_0}{\zeta}, \end{aligned} \tag{A.37}$$

which further gives us

$$\begin{aligned} q_0 &= \frac{(C_0 + p_0)[p_{00}(1 - \zeta) + p_{11}\zeta + p_0]}{p_{11}\zeta + p_{00}(1 - \zeta) + p_0 + C_0 - m_{11}\zeta - m_{00}(1 - \zeta)}, \\ q_{00} &= \frac{(p_{00} - m_{00})[p_{00}(1 - \zeta) + p_{11}\zeta + p_0]}{p_{11}\zeta + p_{00}(1 - \zeta) + p_0 + C_0 - m_{11}\zeta - m_{00}(1 - \zeta)}, \\ q_{11} &= \frac{(p_{11} - m_{11})[p_{00}(1 - \zeta) + p_{11}\zeta + p_0]}{p_{11}\zeta + p_{00}(1 - \zeta) + p_0 + C_0 - m_{11}\zeta - m_{00}(1 - \zeta)}. \end{aligned} \tag{A.38}$$

Define

$$P := p_{00}(1 - \zeta) + p_{11}\zeta + p_0 \quad \text{and} \quad Q := P + C_0 - m_{11}\zeta - m_{00}(1 - \zeta). \tag{A.39}$$

Hence,

$$q_0 = (C_0 + p_0)\frac{P}{Q}, \quad q_{11} = (p_{11} - m_{11})\frac{P}{Q}, \quad q_{00} = (p_{00} - m_{00})\frac{P}{Q}. \tag{A.40}$$

Now using sufficient conditions, we have $p_0 = 0$. The final solution is the following:

$$q_0 = C_0\frac{P}{Q}, \quad q_{11} = (p_{11} - m_{11})\frac{P}{Q}, \quad q_{00} = (p_{00} - m_{00})\frac{P}{Q}, \tag{A.41}$$

where $P := p_{11}\zeta + p_{00}(1-\zeta)$ and $Q := P + C_0 - m_{11}\zeta - m_{00}(1-\zeta)$. We have taken $p_{11} + p_{00} = 1$, but the original $p'_{11} + p'_{00} = \frac{1}{\alpha}$. Therefore, we learn $\widehat{\phi}(C)$ such that such that $\widehat{\phi}(C) = \alpha\phi(C)$. QED.

Corollary A.1. For F_β-measure, where β is unknown, Algorithm 3.1 elicits the true performance metric up to a constant in $O(\log(\frac{1}{\epsilon}))$ queries to the oracle.

Proof. Algorithm 3.1 gives us the supporting hyperplane, the trade-off, and the Bayes confusion matrix. If we know p_{11}, then we can use Proposition 3.4 to compute the other coefficients. In F_β-measure, $p_{11} = 1$, and we do not require Algorithms 3.2 and 3.3. QED.

Proof of Theorem 3.1. We prove the points one by one.

(i) As a direct consequence of our representation of the points on the boundary via their supporting hyperplanes (Section 3.2.1), when we search for the maximizer (mimimizer), we also get the associated supporting hyperplane as well.

(ii) By the nature of binary search, we are effectively narrowing our search interval around some target angle θ_0. Furthermore, since the oracle queries are correct unless the ϕ values are within ϵ_Ω, we must have $|\phi(C_{\bar{\theta}}) - \phi(C_{\theta_0})| < \epsilon_\Omega$, and we output θ' such that $|\theta_0 - \theta'| < \epsilon$. Now, we want to check the bound $|\phi(C_{\theta'}) - \phi(C_{\bar{\theta}})|$. In order to do that, we will also consider the threshold corresponding to the supporting hyperplanes at C_θ's, i.e. $\delta_\theta = {}^{\sin\theta}/_{\sin\theta + \cos\theta}$.

Notice that,

$$\begin{aligned} |\phi(C_{\bar{\theta}}) - \phi(C_{\theta'})| &= |\phi(C_{\bar{\theta}}) - \phi(C_{\theta_0}) + \phi(C_{\theta_0}) - \phi(C_{\theta'})| \\ &\leq |\phi(C_{\bar{\theta}}) - \phi(C_{\theta_0})| + |\phi(C_{\theta_0}) - \phi(C_{\theta'})| \end{aligned} \tag{A.42}$$

The first term is bounded by ϵ_Ω due to the oracle assumption. For the bounds the second term, consider the following.

$$|TP(C_{\theta_0}) - TP(C_{\theta'})|$$

$$
\begin{aligned}
&= \left| \int\limits_{x: \frac{sin\theta_0}{sin\theta_0 + cos\theta_0} \geq \eta(x) \geq \frac{sin\theta'}{sin\theta' + cos\theta'}} \eta(x)\, \mathrm{d}f_X \right| \\
&\leq \left| \int\limits_{x: \frac{sin\theta_0}{sin\theta_0 + cos\theta_0} - \bar{\delta} \geq \eta(x) - \bar{\delta} \geq \frac{sin\theta'}{sin\theta' + cos\theta'} - \bar{\delta}} \mathrm{d}f_X \right| \\
&= \left| \int\limits_{x: \frac{sin\theta_0}{sin\theta_0 + cos\theta_0} - \frac{sin\bar{\theta}}{sin\bar{\theta} + cos\bar{\theta}} \geq \eta(x) - \bar{\delta} \geq \frac{sin\theta'}{sin\theta' + cos\theta'} - \frac{sin\bar{\theta}}{sin\bar{\theta} + cos\bar{\theta}}} \mathrm{d}f_X \right| \\
&= \left| \int\limits_{x: \frac{sin(\theta_0 - \bar{\theta})}{sin(\theta_0 + \bar{\theta}) + cos(\theta_0 - \bar{\theta})} \geq \eta(x) - \bar{\delta} \geq \frac{sin\theta'}{sin\theta' + cos\theta'} - \frac{sin\bar{\theta}}{sin\bar{\theta} + cos\bar{\theta}}} \mathrm{d}f_X \right|,
\end{aligned} \tag{A.43}
$$

where the inequality in the second step follows from the fact that $\eta(x) \leq 1$.

Recall that the left term in the integral limits is actually, $\delta_{\theta_0} - \delta_{\bar{\theta}}$. When $|\phi(C_{\delta_{\theta_0}}) - \phi(C_{\delta_{\bar{\theta}}})| < \epsilon_\Omega$, then we have $|\bar{\delta} - \delta_0| < \frac{2}{k_0}\sqrt{k_1 \epsilon_\Omega}$. The proof of this statement is given in the proof of Theorem 3.2 (proved later). Since sin is 1-Lipschitz, adding and subtracting $\sin\theta_0 / (\sin\theta_0 + \cos\theta_0)$ in the right term of the integration limit gives us the minimum value of the right term to be $-\epsilon - \frac{2\sqrt{k_1 \epsilon_\Omega}}{k_0}$. This implies that the quantity in (A.43) is less than

$$
\begin{aligned}
&\mathbb{P}[\{(\eta(X) - \bar{\delta}) \leq \frac{2}{k_0}\sqrt{k_1 \epsilon_\Omega}\} \cap \{(\bar{\delta} - \eta(X)) \leq \epsilon + \frac{2}{k_0}\sqrt{k_1 \epsilon_\Omega}\}] \\
&\leq \mathbb{P}[(\bar{\delta} - \eta(X)) \leq \epsilon + \frac{2}{k_0}\sqrt{k_1 \epsilon_\Omega}] \\
&\leq \frac{2k_1}{k_0}\sqrt{k_1 \epsilon_\Omega} + k_1 \epsilon. \quad \text{(by Assumption 3.4)}
\end{aligned} \tag{A.44}
$$

As $\mathbb{P}(A \cap B) \leq min\{\mathbb{P}(A), \mathbb{P}(B)\}$, the inequality used in the second step is rather loose, but it shows the dependency on sufficiently small ϵ. It could be independent of the tolerance ϵ depending on the $\mathbb{P}(\eta(X) - \bar{\delta})$ or the sheer big value of ϵ. Nevertheless, a similar result applies to the true negative rate. Since ϕ is 1-Lipschitz, we have that $|\phi(C) - \phi(C')| \leq 1 \cdot \|C - C'\|$, but

$$
\|C(\theta_0) - C(\theta')\|_\infty \leq \frac{2k_1}{k_0}\sqrt{k_1 \epsilon_\Omega} + k_1 \epsilon. \tag{A.45}
$$

Hence,

$$|\phi(C_{\theta'}) - \phi(C_{\overline{\theta}})| \leq \sqrt{2}(\frac{2k_1}{k_0}\sqrt{k_1\epsilon_\Omega} + k_1\epsilon) + \epsilon_\Omega. \tag{A.46}$$

Since the metrics are in $[0,1]$, $\epsilon_\Omega \in [0,1]$. Therefore, $\sqrt{\epsilon_\Omega} \geq \epsilon_\Omega$. This gives us the desired result.

(iii) We needed only, for part (ii), that the interval of possible values of θ' be at most ϵ to the target angle θ_0. Ideally, this is obtained by making $\log_2(1/\epsilon)$ queries, but due to the region where oracle misreport its preferences, we can be off to the target angle θ_0 by more than ϵ.

However, binary search will again put us back in the correct direction, once we leave the misreporting region. And this time, even if we are off to the target angle θ_0, we will be closer than before. Therefore, for the interval of possible values of θ' to be at most ϵ, we require at least $\log(\frac{1}{\epsilon})$ rounds of the algorithm, each of which is a constant number of pairwise queries.

QED.

Proof of Lemma 3.2. For any fixed ϵ, divide the search space θ into bins of length ϵ, resulting in $\lceil \frac{1}{\epsilon} \rceil$ classifiers. When the function evaluated on these classifiers is unimodal, and when the only operation allowed is pairwise comparison, the optimal worst case complexity for finding the argument maximum (of function evaluations) is $O(\log \frac{1}{\epsilon})$ [130], which is achieved by binary search. QED.

Proposition A.1. Let $(y_1, x_1, h(x_1))$, …, $(y_n, x_n, h(x_n))$ be n i.i.d. samples from the joint distribution on Y, X, and $h(X)$. Then by Höffding's inequality,

$$\mathbb{P}\left[\left|\tfrac{1}{n}\textstyle\sum_{i=1}^{n} \mathbf{1}[h_i = y_i = 1] - TP(h)\right| \geq \epsilon\right] \leq 2e^{-2n\epsilon^2}. \tag{A.47}$$

The same holds for the analogous estimator on TN.

Proof. Direct application of Höffding's inequality. QED.

Proof of Theorem 3.2. We will show this for threshold classifiers, as in the statement of the Assumption 3.4, but it is not difficult to extend the argument to the case of querying angles. (Involves a good bit of trigonometric identities...)

Recall, the threshold estimator h_δ returns positive if $\eta(x) \geq \delta$, and zero otherwise. Let $\overline{\delta}$ be the threshold which maximizes performance with respect to ϕ, and $C_{\overline{\delta}}$ be its confusion matrix. For simplicity, suppose that $\delta' < \overline{\delta}$. Recall, from Assumption 3.4 that $\Pr[\eta(X) \in$

$[\bar{\delta} - \frac{k_0}{2k_1}\epsilon,\, \bar{\delta}]] \leq k_0\epsilon/2$, but $\Pr[\eta(X) \in [\bar{\delta} - \epsilon, \bar{\delta}]] \geq k_0\epsilon$, and therefore

$$\mathbb{P}\Big[\eta(X) \in [\bar{\delta} - \epsilon, \bar{\delta} - \tfrac{k_0}{2k_1}\epsilon]\Big] \geq k_0\epsilon/2 \tag{A.48}$$

Denoting $\phi(C) = \langle \mathbf{m}, C\rangle$ and since $\bar{\delta} = m_{00}/(m_{11} + m_{00})$, by expanding the integral, we get

$$\begin{aligned}
&\phi(C_{\bar{\delta}}) - \phi(C_{\delta'}) = \int_{x:\delta' \leq \eta(x) \leq \bar{\delta}} [m_{00}(1 - \eta(x)) - m_{11}\eta(x)]\, \mathrm{d}f_X \\
&= \int_{x:\bar{\delta} - (\bar{\delta} - \delta') \leq \eta(x) \leq \bar{\delta}} [m_{00}(1 - \eta(x)) - m_{11}\eta(x)]\, \mathrm{d}f_X \\
&\geq \int_{x:\bar{\delta} - (\bar{\delta} - \delta') \leq \eta(x) \leq \bar{\delta} - \frac{k_0}{2k_1}(\bar{\delta} - \delta')} [m_{00}(1 - \eta(x)) - m_{11}\eta(x)]\, \mathrm{d}f_X \\
&\geq [(m_{11} + m_{00})\big(\frac{-m_{00}}{m_{00} + m_{11}} + \frac{k_0}{2k_1}(\bar{\delta} - \delta')\big) + m_{00}] \times \int_{x:\bar{\delta} - (\bar{\delta} - \delta') \leq \eta(x) \leq \bar{\delta} - \frac{k_0}{2k_1}(\bar{\delta} - \delta')} \mathrm{d}f_X \\
&= [(m_{11} + m_{00})\frac{k_0}{2k_1}(\bar{\delta} - \delta')] \times \mathbb{P}[\bar{\delta} - (\bar{\delta} - \delta') \leq \eta(x) \leq \bar{\delta} - \frac{k_0}{2k_1}(\bar{\delta} - \delta')] \\
&\geq \tfrac{k_0}{2}(\bar{\delta} - \delta') \cdot \tfrac{k_0}{2k_1}(\bar{\delta} - \delta') = \frac{k_0^2}{4k_1}(\bar{\delta} - \delta')^2.
\end{aligned} \tag{A.49}$$

Similar results hold when $\delta' > \bar{\delta}$. Therefore, if we have $|\phi(\overline{C}) - \phi(C(\delta'))| < \epsilon_\Omega$, then we must have $|\bar{\delta} - \delta'| < \frac{2}{k_0}\sqrt{k_1\epsilon_\Omega}$. Thus, if we are in a regime where the oracle is misreporting the preference ordering, it must be the case that the thresholds are sufficiently close to the optimal threshold.

Again, as in the proof of Theorem 3.1, when the tolerance ϵ is small, our binary search closes in on a parameter θ' which has $\phi(C_{\delta_{\theta'}})$ within ϵ_Ω of the optimum, but from the above discussion, this also implies that the search interval itself is close to the true value, and thus, the total error in the threshold is at most $\epsilon + \frac{2}{k_0}\sqrt{k_1\epsilon_\Omega}$. Since $\bar{\delta} = m_{00}/(m_{11} + m_{00})$, this bound extends to the cost vector with a factor of $\sqrt{2}$, thus giving the desired result.

We observe that the above theorem actually provide bounds on the slope of the hyperplanes. Thus, the guarantees for LFPM elicitation follow naturally. It only requires that we recover the slope at the upper boundary and lower boundary correctly (within some bounds). This theorem provides those guarantees. Algorithm 3.3 is independent of oracle queries and thus can be run with high precision, making the solutions of the two systems match. QED.

Proof of Lemma 3.3. Suppose the performance metric of the oracle is characterized by the parameter $\bar{\theta}$. Recall the Bayes optimal classifier would be $h_{\bar{\theta}} = \mathbf{1}[\eta \geq \bar{\delta}]$. Let us assume we are given a classifier $\widehat{h}_{\bar{\theta}} = \mathbf{1}[\widehat{\eta} \geq \bar{\delta}]$. Notice that the optimal threshold $\bar{\delta}$ is the property of the metric and not the classifier or η. We want to bound the difference in the confusion matrices for these two classifiers. Notice that, by Assumption 3.3, we can take n sufficiently large so that $\|\eta - \widehat{\eta}_n\|_\infty$ is arbitrarily small. Consider the quantity

$$TP(h_{\bar{\theta}}) - TP(\widehat{h}_{\bar{\theta}}) = \int_{\eta \geq \bar{\delta}} \eta \, \mathrm{d} f_X - \int_{\widehat{\eta} \geq \bar{\delta}} \eta \, \mathrm{d} f_X \tag{A.50}$$

Now the maximum loss in the above quantity can occur when, in the region where the classifiers' predictions differ, there $\widehat{\eta}$ is less than η with the maximum possible difference. This is equal to

$$\begin{aligned} &\int\limits_{x : \bar{\delta} \leq \eta(x) \leq \bar{\delta} + \|\eta - \widehat{\eta}\|_\infty} \eta \, \mathrm{d} f_X \\ &\leq \mathbb{P}[\bar{\delta} \leq \eta(X) \leq \bar{\delta} + \|\eta - \widehat{\eta}\|_\infty] \\ &\leq k_1 \|\eta - \widehat{\eta}\|_\infty . \qquad \text{(by Assumpition 3.4)} \end{aligned} \tag{A.51}$$

Similarly, we can look at the maximum gain in the following quantity.

$$TP(\widehat{h}_{\bar{\theta}}) - TP(h_{\bar{\theta}}) = \int_{\widehat{\eta} \geq \bar{\delta}} \eta \, \mathrm{d} f_X - \int_{\eta \geq \bar{\delta}} \eta \, \mathrm{d} f_X \tag{A.52}$$

Now the maximum gain in the above quantity can occur when, in the region where the classifiers' predictions differ, there $\widehat{\eta}$ is greater than η with the maximum possible difference. This is equal to

$$\begin{aligned} \int\limits_{x : \bar{\delta} - \|\eta - \widehat{\eta}\|_\infty \leq \eta(x) \leq \bar{\delta}} \eta \, \mathrm{d} f_X &\leq \mathbb{P}[\bar{\delta} - \|\eta - \widehat{\eta}\|_\infty \\ &\leq \eta(X) \leq \bar{\delta}] \\ &\leq k_1 \|\eta - \widehat{\eta}\|_\infty . \qquad \text{(by Assumpition 3.4)} \end{aligned} \tag{A.53}$$

Hence,

$$|TP(\widehat{h}_{\bar{\theta}}) - TP(h_{\bar{\theta}})| \leq k_1 \|\eta - \widehat{\eta}\|_\infty . \tag{A.54}$$

Similar arguments apply for TN, which gives us the desired result. QED.

A.3 EXTENDED EXPERIMENTS

In this section, we empirically validate the theory and robustness to finite samples.

Table A.1: Empirical Validation for LPM elicitation at tolerance $\epsilon = 0.02$ radians. ϕ^* and $\widehat{\phi}$ denote the true and the elicited metric, respectively.

$\phi^* = \mathbf{m}^*$	$\widehat{\phi} = \widehat{\mathbf{m}}$	$\phi^* = \mathbf{m}^*$	$\widehat{\phi} = \widehat{\mathbf{m}}$
(0.98,0.17)	(0.99,0.17)	(-0.94,-0.34)	(-0.94,-0.34)
(0.87,0.50)	(0.87,0.50)	(-0.77,-0.64)	(-0.77,-0.64)
(0.64,0.77)	(0.64,0.77)	(-0.50,-0.87)	(-0.50,-0.87)
(0.34,0.94)	(0.34,0.94)	(-0.17,-0.98)	(-0.17,-0.99)

A.3.1 Synthetic Data Experiments

We take the same distribution as in (A.1) with the noise parameter $a = 5$. In the LPM elicitation case, we define a true metric ϕ^* by $\mathbf{m}^* = (m^*_{11}, m^*_{00})$. This defines the query outputs in line 6 of Algorithm 3.1. Then we run Algorithm 3.1 to check whether or not we get the same metric. The results for both monotonically increasing and monotonically decreasing LPM are shown in Table A.1. We achieve the true metric even for very tight tolerance $\epsilon = 0.02$ radians.

Next, we elicit LFPM. We define a true metric ϕ^* by $\{(p^*_{11}, p^*_{00}), (q^*_{11}, q^*_{00}, q^*_0)\}$. Then, we run Algorithm 3.1 with $\epsilon = 0.05$ to find the hyperplane $\bar{\ell}$ and maximizer on ∂C_+, Algorithm 3.2 with $\epsilon = 0.05$ to find the hyperplane $\underline{\ell}$ and minimizer on ∂C_-, and Algorithm 3.3 with $n = 2000$ (1000 confusion matrices on both $\partial\mathcal{C}_+$ and $\partial\mathcal{C}_-$ obtained by varying parameter θ uniformly in $[0, \pi/2]$ and $[\pi, 3\pi/2]$) and $\Delta = 0.01$. This gives us the elicited metric $\widehat{\phi}$, which we represent by $\{(\widehat{p}_{11}, \widehat{p}_{00}), (\widehat{q}_{11}, \widehat{q}_{00}, \widehat{q}_0)\}$. In Table A.2, we present the elicitation results for LFPMs (column 2). We also present the mean (α) and the standard deviation (σ) of the ratio of the elicited metric $\widehat{\phi}$ to the true metric ϕ over the set of confusion matrices (column 3 and 4 of Table A.2). As suggested in Corollary A.1, if we know the true ratio of p^*_{11}/p^*_{00}, then we can elicit the LFPM up to a constant by only using Algorithm 3.1 resulting in better estimate of the true metric, because we avoid errors due to Algorithms 3.2 and 3.3. Line 1 and line 2 of Table A.2 represent F_1 measure and $F_{\frac{1}{2}}$ measure, respectively. In both the cases, we assume the knowledge of $p^*_{11} = 1$. Line 3 to line 6 correspond to some arbitrarily chosen linear fractional metrics to show the efficacy of the proposed method. For a better judgment, we show function evaluations of the true metric and the elicited metric on selected pairs of $(TP, TN) \in \partial\mathcal{C}_+$ (used for Algorithm 3.3) in Figure A.2. The true and the elicited metric are plotted together after sorting values based on slope parameter θ. We see that the elicited metric is a constant multiple of the true metric. The vertical solid and dashed line corresponds to the *argmax* of the true and the elicited metric, respectively. In Figure A.2, we see that the *argmax* of the true and elicited metrics coincides, thus validating Theorem 3.1.

Table A.2: LFPM Elicitation for synthetic distribution (Section A.3.1) and Magic (M) dataset (Section A.3.2) with $\epsilon = 0.05$ radians. $(p^*_{11}, p^*_{00}), (q^*_{11}, q^*_{00}, q^*_0)$ denote the true LFPM. $(\widehat{p}_{11}, \widehat{p}_{00}), (\widehat{q}_{11}, \widehat{q}_{00}, \widehat{q}_0)$ denote the elicited LFPM. α and σ denote the mean and the standard deviation in the ratio of the elicited to the true metric (evaluated on the confusion matrices in $\partial\mathcal{C}_+$ used in Algorithm 3.3), respectively. We empirically verify that the elicited metric is constant multiple (α) of the true metric.

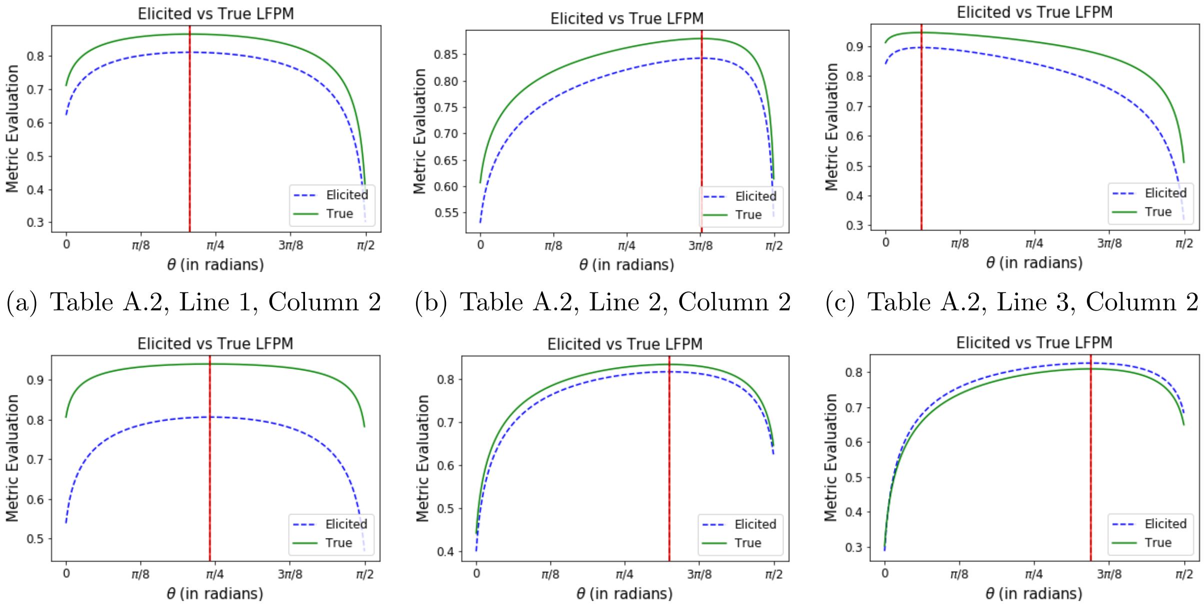
True Metric	Results on Synthetic Distribution (Section A.3.1)			Results on Real World Dataset M (Section A.3.2)		
$(p^*_{11}, p^*_{00}), (q^*_{11}, q^*_{00}, q^*_0)$	$(\widehat{p}_{11}, \widehat{p}_{00}), (\widehat{q}_{11}, \widehat{q}_{00}, \widehat{q}_0)$	α	σ	$(\widehat{p}_{11}, \widehat{p}_{00}), (\widehat{q}_{11}, \widehat{q}_{00}, \widehat{q}_0)$	α	σ
(1.00,0.00),(0.50,-0.50,0.50)	(1.00,0.00),(0.25,-0.75,0.75)	0.92	0.03	(1.00,0.00),(0.25,-0.75,0.75)	0.90	0.06
(1.0,0.0),(0.8,-0.8,0.5)	(1.0,0.0),(0.73,-1.09,0.68)	0.94	0.02	(1.0,0.0),(0.72,-1.13, 0.57)	1.06	0.05
(0.8,0.2),(0.3,0.1,0.3)	(0.86,0.14),(-0.13,-0.07, 0.60)	0.90	0.06	(0.23,0.77),(-0.87,0.66,0.76)	0.84	0.09
(0.60,0.40),(0.40,0.20,0.20)	(0.67,0.33),(-0.07,-0.44,76)	0.82	0.05	(0.16,0.84),(-0.89,0.25,0.89)	0.65	0.05
(0.40,0.60),(-0.10,-0.20,0.65)	(0.36,0.64),(-0.21,-0.25,0.73)	0.97	0.01	(0.08,0.92),(-0.75,0.12,0.82)	0.79	0.08
(0.20,0.80),(-0.40,-0.20,0.80)	(0.12, 0.88),(-0.43, 0.002, 0.71)	1.02	0.006	(0.19,0.81),(-0.38,-0.13,0.70)	1.02	0.004

A.3.2 Real-World Data Experiments

In real-world datasets, we do not know $\eta(x)$ and only have finite samples. Thus, the feasible space $\mathcal{C}$ is not as well behaved as shown in Figure A.1, and poses a challenge for the elicitation task. Now, we validate the elicitation procedure with two real-world datasets. The datasets are: (a) Breast Cancer (BC) Wisconsin Diagnostic dataset [28] containing 569 instances, and (b) Magic (M) dataset [29] containing 19020 instances. For both the datasets, we standardize the attributes and split the data into two parts $\mathcal{S}_1$ and $\mathcal{S}_2$. On $\mathcal{S}_1$, we learn an estimator $\widehat{\eta}$ using regularized logistic regression model with regularizing constant $\lambda = 10$ and $\lambda = 1$. We use $\mathcal{S}_2$ for making predictions and computing sample confusions.

We generated twenty eight different LPMs ϕ^* by generating θ^* (or say, $\mathbf{m}^* = (\cos\theta^*, \sin\theta^*)$). Fourteen from the first quadrant starting from $\pi/18$ radians to $5\pi/12$ radians in step of $\pi/36$ radians. Similarly, fourteen from the third quadrant starting from $19\pi/18$ to $17\pi/12$ in step of $\pi/36$ radians. We then use Algorithm 3.1 (Algorithm 3.2) for different tolerance ϵ, for different datasets, and for different regularizing constant λ in order to recover the estimate $\widehat{\mathbf{m}}$. We compute the error in terms of the proportion of the number of times when Algorithm 3.1 (Algorithm 3.2) failed to recover the true $\mathbf{m}^*$ within ϵ threshold.

We report our results in Table A.3. We see improved elicitation for dataset M, suggesting that ME improves with larger datasets. In particular, for dataset M, we elicit all the metrics within threshold $\epsilon = 0.11$ radians. We also observe that $\epsilon = 0.02$ is an overly tight tolerance for both the datasets leading to many failures. This is because the elicitation routine gets stuck at the closest achievable confusion matrix from finite samples, which need not be optimal within the given (small) tolerance. Furthermore, both of these observations are consistent for both the regularized logisitic regression models with regularizer λ.



(a) Table A.2, Line 1, Column 2 (b) Table A.2, Line 2, Column 2 (c) Table A.2, Line 3, Column 2

(d) Table A.2, Line 4, Column 2 (e) Table A.2, Line 5, Column 2 (f) Table A.2, Line 6, Column 2

Figure A.2: True and elicited LFPMs for synthetic distribution from Table A.2. The solid green curve and the dashed blue curve are the true and the elicited metric, respectively. The solid red and the dashed black vertical lines represent the maximizer of the true metric and the elicited metric, respectively. The elicited LFPMs are constant multiple of the true metrics with the same maximizer (solid red and dashed black vertical lines overlap).

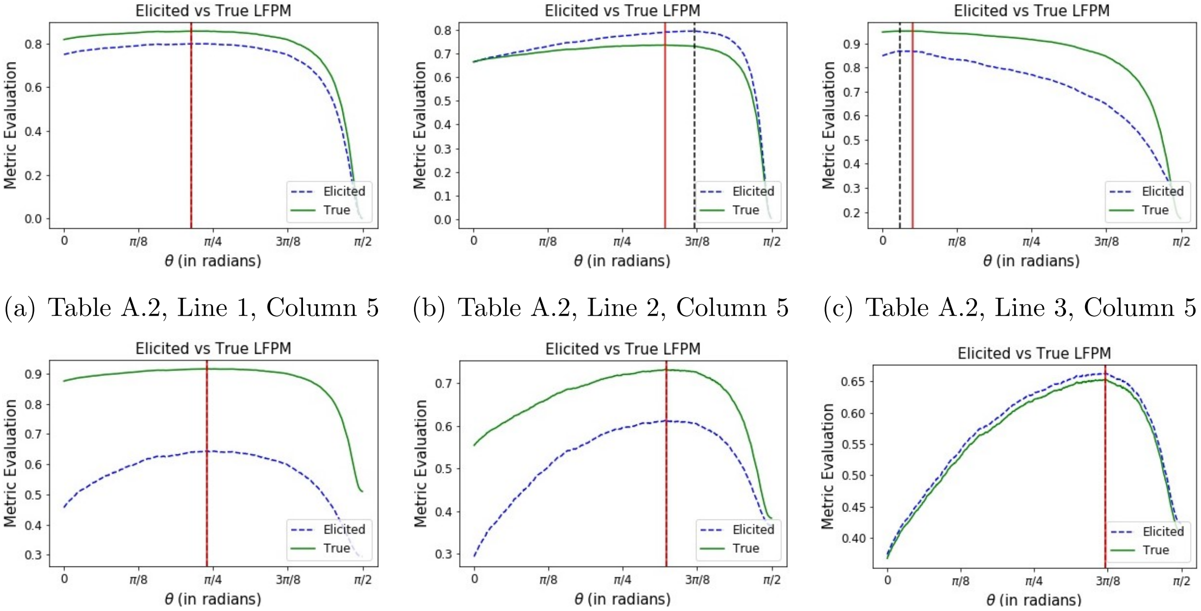
Next, we discuss the case of LFPM elicitation. We use the same true metrics ϕ^* as described in Section A.3.1 and follow the same process for eliciting LFPM, but this time we work with MAGIC dataset. In Table A.2 (columns 5, 6, and 7), we present the elicitation results on MAGIC dataset along with the mean α and the standard deviation σ of the ratio of the elicited metric and the true metric. Again, for a better judgment, we show the function evaluation of the true metric and the elicited metric on the selected pairs of $(TP, TN) \in \partial\mathcal{C}_+$ (used for Algorithm 3.3) in Figure A.3, ordered by the parameter θ. Although we do observe that the *argmax* is different in two out of six cases (see Sub-figure (b) and Sub-figure (c)) due to finite samples, elicited LFPMs are almost equivalent to the true metric up to a constant.

A.4 MONOTONICALLY DECREASING CASE

If the oracle's metric is monotonically decreasing in TP, TN, we can find the supporting hyperplanes at the maximizer and the minimizer. It would require to pose one query $\Omega(C^*_{\pi/4}, C^*_{5\pi/4})$. The response determines whether we want to search over $\partial\mathcal{C}_+$ or $\partial\mathcal{C}_-$ and apply Algorithms 3.1 and 3.2 accordingly. If $C^*_{\pi/4} \prec C^*_{5\pi/4}$, then the metric is monotonically decreasing, and we search for the maximizer on the lower boundary $\partial\mathcal{C}_-$ (and vice-versa).

Table A.3: LPM elicitation results on real datasets (ϵ in radians). M and BC represent Magic and Breast Cancer dataset, respectively. λ is the regularization parameter in the regularized logistic regression models. The table shows error in terms of the proportion of the number of times when Algorithm 3.1 (Algorithm 3.2) failed to recover the true $\mathbf{m}^*(\theta^*)$ within ϵ threshold. The observations made in Chapter 3 are consistent for both models.

	$\lambda = 10$		$\lambda = 1$	
ϵ	M	BC	M	BC
0.02	0.57	0.79	0.54	0.79
0.05	0.14	0.43	0.36	0.64
0.08	0.07	0.21	0.14	0.57
0.11	0.00	0.07	0.07	0.43



(a) Table A.2, Line 1, Column 5 (b) Table A.2, Line 2, Column 5 (c) Table A.2, Line 3, Column 5

(d) Table A.2, Line 4, Column 5 (e) Table A.2, Line 5, Column 5 (f) Table A.2, Line 6, Column 5

Figure A.3: True and elicited LFPMs for dataset M from Table A.2. The solid green curve and the dashed blue curve are the true and the elicited metric, respectively. The solid red and the dashed black vertical lines represent the maximizer of the true metric and the elicited metric, respectively. We see that the elicited LFPMs are constant multiple of the true metrics with almost the same maximizer (solid red and dashed black vertical lines overlap except for two cases).

APPENDIX B: MULTICLASS CLASSIFICATION PERFORMANCE METRIC ELICITATION

Let f_X be the marginal distribution for $\mathcal{X}$.

B.1 SHRINKINTERVAL-1 AND SHRINKINTERVAL-2 SUBROUTINES

Notice that both *ShrinkInterval* sub-routines work with responses to four queries, and based on the responses divides the interval into two. Since the metric dealt in Algorithm 4.1 is concave and unimodal (see Lemma 4.2 and Remark 4.1), four queries are required to shrink the interval into by half in every iteration. Since we use the enclosed sphere for LPM elicitation, we can shrink the interval into half based on just two queries in Algorithm 4.2, i.e. by querying $\Omega(\bar{\mathbf{c}}^d, \bar{\mathbf{c}}^c)$ and $\Omega(\bar{\mathbf{c}}^e, \bar{\mathbf{c}}^d)$, due to strong convexity of the sphere (see proof of Theorem 4.2). However, we show use of four queries in Algorithm 4.2 just to make the algorithms consistent for the readers to understand.

B.2 PROOFS OF SECTION 4.2

Proof of Proposition 4.1. The following are the properties of $\mathcal{D}$.

- *Convex*: Let us take two classifiers $h_1, h_2 \in \mathcal{H}$ which achieve the diagonal confusions $\mathbf{d}(h_1), \mathbf{d}(h_2) \in \mathcal{D}$. We need to check whether there exists a classifier, which achieves the off-diagonal confusion $\lambda\mathbf{d}(h_1)+(1-\lambda)\mathbf{d}(h_2)$. Consider a classifier h, which with probability λ predicts what classifier h_1 predicts and with probability $1-\lambda$ predicts what classifier h_2 predicts. Then the first component

$$\begin{aligned} d_1(h) &= \mathbb{P}(Y=1, h=1) \\ &= \mathbb{P}(Y=1, h=h_1 | h=h_1)\mathbb{P}(h=h_1) + \mathbb{P}(Y=1, h=h_2 | h=h_2)\mathbb{P}(h=h_2) \\ &= \lambda d_1(h_1) + (1-\lambda) d_1(h_2). \end{aligned} \tag{B.1}$$

  Similarly, this hold true for $d_i(h)$ for $i \in [k]$. Hence, C is convex.

- *Bounded:* Since $D_i = P[Y=i, h=i] \leq \zeta_i$ for all $i \in [K]$, $\mathcal{D} \subseteq [0, \zeta_1] \times \cdots \times [0, \zeta_k]$.

- *Strictly convex and closed:* Since $\mathcal{C}$ is convex, its boundary is intersection of half spaces. Furthermore, any linear functional is maximized at the boundary of a convex set [26]. Suppose we are given a diagonal linear functional (DLPM) $\mathbf{a}$. The BO classifier $h^{\mathbf{a}}$ for

```
Subroutine ShrinkInterval-1
Input: Oracle responses for $\Omega(\overline{\mathbf{d}}^c_{1,i}, \overline{\mathbf{d}}^a_{1,i})$,
        $\Omega(\overline{\mathbf{d}}^d_{1,i}, \overline{\mathbf{d}}^c_{1,i}), \Omega(\overline{\mathbf{d}}^e_{1,i}, \overline{\mathbf{d}}^d_{1,i}), \Omega(\overline{\mathbf{d}}^b_{1,i}, \overline{\mathbf{d}}^e_{1,i})$.
If  $(\overline{\mathbf{d}}^a_{1,i} \succ \overline{\mathbf{d}}^c_{1,i})$ $m^b = m^d$.
elseif  $(\overline{\mathbf{d}}^a_{1,i} \prec \overline{\mathbf{d}}^c_{1,i} \succ \overline{\mathbf{d}}^d_{1,i})$ $m^b = m^d$.
elseif  $(\overline{\mathbf{d}}^c_{1,i} \prec \overline{\mathbf{d}}^d_{1,i} \succ \overline{\mathbf{d}}^e_{1,i})$ $m^a = m^c$, $m^b = m^e$.
elseif  $(\overline{\mathbf{d}}^d_{1,i} \prec \overline{\mathbf{d}}^e_{1,i} \succ \overline{\mathbf{d}}^b_{1,i})$ $m^a = m^d$.
else $m^a = m^d$.
Output: $[m^a, m^b]$.
```

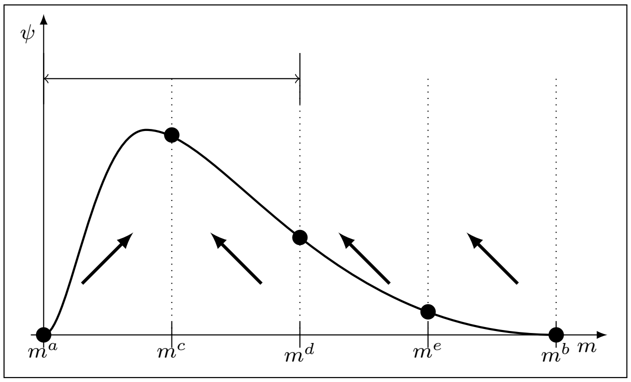


Figure B.1: (Left): Description of Subroutine *ShrinkInterval-1*. (Right): Visual intuition of the subroutine *ShrinkInterval-1*; in search of the maximizer of a quasiconcave metric ψ, the subroutine shrinks the current interval to half based on oracle responses to the four queries.

```
Subroutine ShrinkInterval-2
Input: Oracle responses for $\Omega(\overline{\mathbf{c}}^c, \overline{\mathbf{c}}^a), \Omega(\overline{\mathbf{c}}^d, \overline{\mathbf{c}}^c)$,
        $\Omega(\overline{\mathbf{c}}^e, \overline{\mathbf{c}}^d), \Omega(\overline{\mathbf{c}}^b, \overline{\mathbf{c}}^e), j \in [q]$.
If  $(\overline{\mathbf{c}}^a \succ \overline{\mathbf{c}}^c)$ $\theta^b_j = \theta^d_j$.
elseif  $(\overline{\mathbf{c}}^a \prec \overline{\mathbf{c}}^c \succ \overline{\mathbf{c}}^d)$ $\theta^b_j = \theta^d_j$.
elseif  $(\overline{\mathbf{c}}^c \prec \overline{\mathbf{c}}^d \succ \overline{\mathbf{c}}^e)$ $\theta^a_j = \theta^c_j$, $\theta^b_j = \theta^e_j$.
elseif  $(\overline{\mathbf{c}}^d \prec \overline{\mathbf{c}}^e \succ \overline{\mathbf{c}}^b)$ $\theta^a_j = \theta^d_j$.
else Set $\theta^a_j = \theta^d_j$.
Output: $[\theta^a_j, \theta^b_j]$.
```

Figure B.2: Formal description of the subroutine *ShrinkInterval-2*. *ShrinkInterval-2* is same as *ShrinkInterval-1* except that it applies to the parameter θ_j and works with responses to off-diagonal confusions based queries.

that function is given by Proposition B.1 (whose proof is discussed later). Let the value achieved by the corresponding *BO* diagonal confusion $\overline{d}$ is α. That is,

$$\alpha = \sum_{i=1}^{k} a_i d_i = \sum_{i=1}^{k} \int_{\mathcal{X}} a_i \eta_i(\mathbf{x}) \mathbf{1}[h^{\mathbf{a}}(\mathbf{x}) = i | X = \mathbf{x}] df_X. \tag{B.2}$$

Now, if we want to construct another classifier which achieves the same value α, there has to be some weight shift from one class to another class without changing the maximum value α, but note that $\mathbb{P}[a_i\eta_i(X) = a_j\eta_j(X)] = 0$ for all $i, j \in [k]$ due to Assumption 4.1.

Hence, there is a unique maximizer of this linear functional on the boundary. Therefore, the space is strictly convex. One characterization of the boundary of the space $\partial\mathcal{D}$ can be given by BO diagonal-confusions corresponding to any linear functional $\mathbf{a}$. These diagonal confusions are achieved by the corresponding BO classifiers. Therefore, these diagonal confusions are always achievable, and the space is closed as well.

- $\mathbf{v}_i$ *are always achieved:* It is easy to see that any trivial classifier which predicts only class $i \in [k]$, will achieve the diagonal confusion defined by $\mathbf{v}_i$.

- $\mathbf{v}_i$ *are the only vertices:* Certainly, a vertex exists if (and only if) some point is supported by more than k tangent hyperplanes in k dimensional space. This means that the vertex is optimal for more than k linear metric (linear functional). Clearly, all the metrics with slope $\mathbf{a}$ such that $a_i > a_j > 0$ and $a_l = 0 \; \forall \; l \in [k], l \neq i, j$ support $\mathbf{v}_i$. So, there are at least k supporting hyperplanes at these points, which make them the vertices. Now, we show that these are the only vertices.

  Suppose there is a point other than $\mathbf{v}_i$'s which is supported by two hyperplanes given by the slopes $\mathbf{a}^1$ and $\mathbf{a}^2$. From Proposition B.1 (discussed later), we can get Bayes optimal classifiers $h^{\mathbf{a}^1}$ and $h^{\mathbf{a}^2}$, which achieve the same diagonal confusions. This means that

$$\int_{\mathbf{x}: \frac{\eta_1(\mathbf{x})}{\eta_j(\mathbf{x})} \geq t_j, j \in \{2, \cdots, K\}} \eta_1(\mathbf{x}) df_X = \int_{\mathbf{x}: \frac{\eta_1(\mathbf{x})}{\eta_j(\mathbf{x})} \geq t'_j, j \in \{2, \cdots, K\}} \eta_1(\mathbf{x}) df_X, \tag{B.3}$$

  i.e., the first component d_1 should be equal for the two classifiers, where t_j, t'_j's are dependent on $\mathbf{a}^1$ and $\mathbf{a}^2$. Since, these classifiers are different at least for one j, $t_j \neq t'_j$. This will mean that there are multiple values of $\frac{\eta_1(\mathbf{x})}{\eta_j(\mathbf{x})}$ which are not attained. This contradict with our Assumption 4.1 that g_{1j} is strictly decreasing. By strict convexity, there are no supporting hyperplane tangent at multiple points. Hence, $\mathbf{v}_i$ are the only vertices of the set $\mathcal{D}$.

Since we take classifiers which predict only classes k_1 and k_2, the values of any diagonal confusion $\mathbf{d} \in \mathcal{D}_{k_1,k_2}$ evaluate to zero at indices except k_1, k_2. Therefore, the properties of the space $\mathcal{D}_{k_1,k_2}$ can be proved on similar lines to Chapter 3. QED.

Proof of Proposition 4.3. The following are the properties of the space $\mathcal{C}$.

- *Convex* The space is convex follows from first point of Proposition 4.1.
- *Bounded:* $C_{ij} = \mathbb{P}[Y = i, h = j] \leq \mathbb{P}[Y = i] = \zeta_i$ for $i, j \in [k]$. When confusion matrices written in row major form excluding the diagonal terms, then it is easy to see that $\mathcal{C} \subseteq [0, \zeta_1]^{(k-1)} \times [0, \zeta_2]^{(k-1)} \times \cdots \times [0, \zeta_k]^{(k-1)}$.
- $\mathbf{u}_i$*'s and* $\mathbf{o}$ *are always achieved:* The classifier which always predicts class i, will achieve the confusion matrix $\mathbf{u}_i$. Thus, $\mathbf{u}_i \in \mathcal{C} \, \forall \, i \in [q]$. Furthermore, a classifier which predicts similar to one of the trivial classifiers with probability $1/k$ will achieve the confusions $\mathbf{o}$ (the centroid).

- $\mathbf{u}_i$*'s are vertices:* Any supporting hyperplane with slope $a_{1i} < a_{1j} < 0$ and $a_{1l} = 0$ for $l \in [k], l \neq i, j$ will be supported by $\mathbf{u}_1$ (corresponding to BO classifier which predict class 1). Thus, $\mathbf{u}_1$ is supported by at least q hyperplanes. Thus, it becomes a vertex of the convex set. Similar is the case with other $\mathbf{u}_i$'s.

QED.

Proposition 4.2 can be considered as a corollary of the following more general Proposition.

Proposition B.1. Let $\psi \in \varphi_{DLPM}$, parametrized by $\mathbf{a}$, then

$$\overline{h}(\mathbf{x}) = \operatorname*{argmax}_{i\in[k]} a_i\eta_i(\mathbf{x}), \quad \text{and} \quad \underline{h}(\mathbf{x}) = \operatorname*{argmin}_{i\in[k]} a_i\eta_i(\mathbf{x}) \tag{B.4}$$

are the BO and IBO classifiers *w.r.t* ψ, respectively.

Proof. Let

$$\psi = \sum_i a_i d_i = \sum_i \int_{\mathcal{X}} a_i\eta_i(\mathbf{x})\mathbf{1}[h(\mathbf{x}) = i]. \tag{B.5}$$

From this mathematical form, it is easy to see that the metric achieves its maximum when a class that maximizes the expected utility conditioned on the instance is predicted. That is, the metric achieves its maximum when a classifier deterministically predicts class i when $i = \operatorname{argmax}_{j\in[k]} a_j\eta_j(x)$. This is the form of the classifier written in the proposition. Similarly, this metric is minimized when when a classifier minimizes the expected utility conditioned on the instance, by predicting class $i = \operatorname{argmin}_{j\in[k]} a_j\eta_j(x)$. QED.

Proof of Proposition 4.2. Recall that classifiers which predict only class k_1 and k_2 will achieve diagonal confusions, which have zeros at every other index except k_1, k_2. Therefore,

$$\begin{aligned}\psi &= \sum_i a_i d_i = a_{k_1}d_{k_1} + a_{k_2}d_{k_2} \\ &= \int_{\mathcal{X}} a_{k_1}\eta_{k_1}(x)\mathbf{1}[h(x) = k_1] + \int_{\mathcal{X}} a_{k_2}\eta_{k_2}(x)\mathbf{1}[h(x) = k_2].\end{aligned} \tag{B.6}$$

Again, using the idea used in the previous proof, the metric achieves its maximum when a class that maximizes the expected utility conditioned on the instance is predicted. Therefore,

$$\overline{h}_{k_1,k_2}(x) = \left\{ \begin{array}{ll} k_1, & \text{if } a_{k_1}\eta_{k_1}(\mathbf{x}) \geq a_{k_2}\eta_{k_2}(\mathbf{x}) \\ k_2, & \hfill o.w. \end{array} \right\} \tag{B.7}$$

is the RBO classifier (restricted to classes k_1, k_2) with respect to ψ. Furthermore, the RIBO classifier is given by $\underline{h}_{k_1,k_2}(\mathbf{x}) = k_2 \mathbf{1}[\overline{h}_{k1,k_2}(\mathbf{x}) = k_1] + k_1 \mathbf{1}[\overline{h}_{k1,k_2}(\mathbf{x}) = k_2]$. RIBO classifier does exactly the opposite of RBO, i.e., it predicts class k_1, wherever RBO predicts class k_2 on the instance space $\mathcal{X}$ and vice-versa. QED.

Proof of Lemma 4.1. Suppose the origin is at $\mathbf{o}$ and the constrained set is the sphere $\mathcal{S}_\lambda$ with radius λ centered at $\mathbf{o}$. We want to maximize $\langle \mathbf{a}, \mathbf{c} \rangle$ such that $\mathbf{c} \in \mathcal{S}_\lambda$. Since a linear metric over a convex set is maximized at the boundary [26], it is easy to see that $c_i = \lambda a_i$ will maximize this metric. Moving the reference point to the original origin i.e. $\mathbf{0}^q$ gives us the required answer. QED.

B.3 PROOFS OF SECTION 4.3

We write Lemma 4.2 in the following more general form.

Lemma B.1. Let $\psi : \mathcal{D} \to \mathbb{R}\,(\xi : \mathcal{D} \to \mathbb{R})$ be a quasiconcave (quasiconvex) function, which is monotone increasing in all $\{d_i\}_{i=1}^k$. For $k_1, k_2 \in [k]$, let $\rho^+ : [0,1] \to \partial\mathcal{D}^+_{k_1,k_2}$ ($\rho^- : [0,1] \to \partial\mathcal{D}^-_{k_1,k_2}$) be a continuous, bijective, parametrization of the upper (lower) boundary. Then the composition $\psi \circ \rho^+ : [0,1] \to \mathbb{R}$ ($\xi \circ \rho^- : [0,1] \to \mathbb{R}$) is quasiconcave (quasiconvex) and thus unimodal on the interval $[0,1]$.

Proof. A function is quasiconcave iff super-level sets are convex. We already know from Proposition 4.1 $\mathcal{D}_{k_1,k_2}$ is convex. Moreover, any vector of diagonal confusions has zeros at every index except at indices k_1, k_2. Let $\psi : \mathcal{D} \to \mathbb{R}$ be a quasiconcave metric, which implies that its super-level sets $\mathcal{L}^{\mathcal{D}}_r(\psi) = \{\mathbf{d} \in \mathcal{D} \ : \ \psi(\mathbf{d}) \geq r\}$ are convex. Now, consider the super-level sets of ψ restricted to the diagonal confusions in $\mathcal{D}_{k_1,k_2}$ i.e. $\mathcal{L}^{\mathcal{D}_{k_1,k_2}}_r(\psi) = \{\mathbf{d} \in \mathcal{D}_{k_1,k_2} \ : \ \psi(\mathbf{d}) \geq r\}$. Take any $\mathbf{d}^1, \mathbf{d}^2 \in \mathcal{L}^{\mathcal{D}_{k_1,k_2}}_r(\psi)$. Since $\mathbf{d}^1, \mathbf{d}^2 \in \mathcal{D}$ as well, they belong to the set $\mathcal{L}^{\mathcal{D}}_r(\psi)$, which is convex. Hence, for $t \in [0,1]$, $t\mathbf{d}^1 + (1-t)\mathbf{d}^2 \in \mathcal{L}^{\mathcal{D}}_r(\psi)$, which implies that $\psi(t\mathbf{d}^1 + (1-t)\mathbf{d}^2) \geq r$. Furthermore, $t\mathbf{d}^1 + (1-t)\mathbf{d}^2 \in \mathcal{D}_{k_1,k_2}$, because $\mathcal{D}_{k_1,k_2}$ is convex. By the above two arguments, we have that $t\mathbf{d}^1 + (1-t)\mathbf{d}^2 \in \mathcal{L}^{\mathcal{D}_{k_1,k_2}}_r(\psi)$. This implies that $\mathcal{L}^{\mathcal{D}_{k_1,k_2}}_r(\psi)$ is convex, and hence ψ restricted to $\mathcal{D}_{k_1,k_2}$ is quasiconcave. The proof analogously follows for quasiconvex metric ξ.

Now, it remains to show that $\psi \circ \rho^+ : [0,1] \to \mathbb{R}$ ($\psi \circ \rho^- : [0,1] \to \mathbb{R}$) is quasiconcave (quasiconvex). This can be proved by readily extending the proof of Lemma 3.1 (Chapter 3) to the diagonal multiclass case. For the sake of completeness, we also provide the proof here.

We will prove the result for $\psi \circ \rho^+$ on $\partial\mathcal{D}^+_{k_1,k_2}$, and the argument for $\xi \circ \rho^-$ on $\partial\mathcal{D}^-_{k_1,k_2}$ is essentially the same. For simplicity, we drop the + symbols in the notation. It is given that

ψ is quasiconcave. Let S be some superlevel set of ψ. We first want to show that for any $r < s < t$, if $\rho(r) \in S$ and $\rho(t) \in S$, then $\rho(s) \in S$. Since ρ is a continuous bijection, due to the geometry of $\mathcal{D}_{k_1,k_2}$, we must have — wlog — $d_{k_1}(\rho(r)) < d_{k_1}(\rho(s)) < d_{k_1}(\rho(t))$, and $d_{k_2}(\rho(r)) > d_{k_2}(\rho(s)) > d_{k_2}(\rho(t))$ (otherwise swap r and t). Since the set $\mathcal{D}_{k_1,k_2}$ is strictly convex and the image of ρ is $\partial\mathcal{D}_{k_1,k_2}$, then $\rho(s)$ must dominate (component-wise) a point in the convex combination of $\rho(r)$ and $\rho(t)$. Say that point is z. Since ψ is monotone increasing, then $x \in S \implies y \in S$ for all $y \geq x$ component-wise. Therefore, $\psi(\rho(s)) \geq \psi(z)$. Since, S is convex, $z \in S$ and, due to the argument above, $\rho(s) \in S$.

This implies that $\rho^{-1}(\partial\mathcal{D}_{k_1,k_2} \cap S)$ is an interval, and is therefore convex. Thus, the superlevel sets of $\psi \circ \rho$ are convex, so it is quasiconcave, as desired. This implies unimodaltiy as a function over the real line since a function which has more than one local maximum can not be quasiconcave (consider the super-level set for some value slightly less than the lowest of the two peaks). QED.

B.4 PROOFS OF SECTION 4.5

Proof of Theorem 4.1. In Chapter 3, it is shown that for binary classification, the inner loop of Algorithm 4.1 will estimate the value of $\widehat{m}$ for the Bayes-optimal binary classifier corresponding to a linear metric $\mathbf{a}^* = (m^*, 1 - m^*) \in \mathbb{R}^2$, such that $|\widehat{m} - m^*| < \epsilon + \sqrt{\epsilon_\Omega}$ after $O(\log \frac{1}{\epsilon})$ iterations. Now, in the multiclass case, this allows us to argue that, for any $1 \leq i < j \leq k$, we can estimate a value m_{ij} such that $a_i^*/a_j^* = (1 - m_{ij})/m_{ij}$.

For the required guarantees, wlog, we assumed throughout the algorithm that $a_1 \geq a_k/2$ for all k. This is because, if a_1 does not satisy this condition, then we can always choose an index $z \in [k]$ which does satisfy this from the following procedure:

```
Set $z \leftarrow 1$
for $t = 2, 3, \cdots, k$ do
    Compute an estimate $\widehat{m}_{tz}$ of $m_{tz}$.
    if $\widehat{m}_{tz} < \frac{1}{2}$ then $z \leftarrow t$ else do nothing
end for
Output: $z$.
```

Let $\varepsilon = \epsilon + \sqrt{\epsilon_\Omega}$. Now, if $\widehat{m}_{tz} < \frac{1}{2}$, then $a_t^* \geq a_z^* \cdot (\frac{1}{2} - \varepsilon)/(\frac{1}{2} + \varepsilon) = \frac{1-2\varepsilon}{1+2\varepsilon}$. It can be shown that this ratio is at least $1 - 4\varepsilon$. Therefore, if z is the final coordinate output, we must have that $a_z \geq (1 - 4\varepsilon)^k a_t$ for all t. But $(1 - 4\varepsilon)^k \approx e^{-4k\varepsilon}$, and so for ε sufficiently small, we have $a_z \geq a_t/2$ for all t as desired. Now that we have our assumption, we may proceed to show

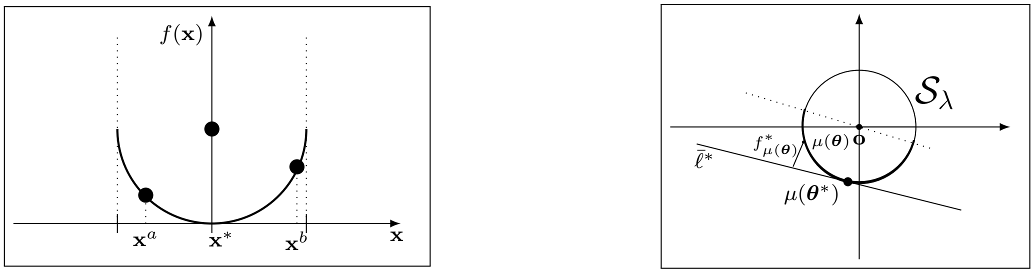


Figure B.3: (Left): A function for the semicircle with unit radius. (Right): Visual intuition for the distance between the boundary points and tangent place at the optimal off-diagonal confusions.

that the algorithm is correct. We wish to show that $\|\widehat{\mathbf{a}}/|\widehat{a}_z| - \mathbf{a}/|a_z|\|_\infty < O(\varepsilon)$. We have

$$\begin{aligned}\left|\frac{\widehat{a}_t}{\widehat{a}_z} - \frac{a_t}{a_z}\right| &= \left|\frac{1-\widehat{m}_t}{\widehat{m}_t} - \frac{1-m_t}{m_t}\right| = \left|\frac{1}{\widehat{m}_t} - \frac{1}{m_t}\right| \\ &\leq \frac{1}{m_t - \varepsilon} - \frac{1}{m_t} \leq \frac{1}{m_t}\left(\frac{1}{1-2\varepsilon} - 1\right) \leq 2 \cdot 2\varepsilon/(1-2\varepsilon) \leq 5\varepsilon \qquad \text{(B.8)}\end{aligned}$$

for $\varepsilon < 0.1$. This gives us the deisred bound. QED.

Proof of Theorem 4.2. Consider the geometry shown in the Figure B.3 (left). This shows a function $f[-1,1]^q \to \mathbb{R}$ which follow the trajectory of a unit semicircle (semisphere). Let $\mathbf{x}$ be a q-dimensional vector, then this function is given by:

$$f(\mathbf{x}) = 1 - \sqrt{1 - \sum_i^q x_i^2}. \qquad \text{(B.9)}$$

Intuitively, this function evaluates the distance of the points lying on the surface of the semisphere. The point $\mathbf{x}^*$ (the origin) is the unique minimizer of this function. Let us restrict the domain of this function to the points $Q = [\mathbf{x}^a, \mathbf{x}^b]$, where $\mathbf{x}^a > -1$ (component-wise) and $\mathbf{x}^b < 1$ (component-wise). Then it is easy to see that the derivative of this function:

$$\nabla f = \left(\frac{x_1}{\sqrt{1-\sum_i^q x_i^2}}, \ldots, \frac{x_q}{\sqrt{1-\sum_i^q x_i^2}}\right) \qquad \text{(B.10)}$$

is continuously differentiable on a compact domain Q. Thus, ∇f is Lipschitz with some Lipschitz parameter L i.e.:

$$\|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x})\|_2 \leq L\|y - x\|_2 \qquad \text{(B.11)}$$

which makes the function f to be L-smooth. In addition, we observe that:

$$f(\mathbf{x}) = 1 - \sqrt{1 - \sum_i^q x_i^2} \geq \frac{1}{2}\sum_i^q x_i^2. \tag{B.12}$$

This implies that there exists a paraboloid always below the function f, which by definition, makes the function f a strongly convex function (say with strong convexity parameter τ). Thus, this function satisfies all the requirements i.e smoothness, strong convexity, and has unique minimizer, to inherit the guarantees from Derivative Free Optimization [36]. Notice that if we apply the coordinate-wise binary search Algorithm 4.2, where the inner loop is run for $\log(1/\epsilon)$ queries, to minimize this function using pairwise comparison queries (i.e. the oracle responds with the point that evaluate to lesser value of f out of the two), then by Theorem 5 of [36] one can guarantee that after $\frac{4L}{\tau}\log(\frac{f(\mathbf{x}^0)-f(\mathbf{x}^*)}{\epsilon^2 2qL^2/\tau})q\log(1/\epsilon)$ queries to the oracle, we can get an estimate of the minimizer $\mathbf{x}^T$ such that $f(\mathbf{x}^T) - f(\mathbf{x}^*) < 4qL^2\epsilon^2/\tau$. Notice that for this function $f(x^0) - f(x^*) = f(x^0) - 0 = f(x^0) \leq 1$.

Now, for simplicity assume $\lambda = 1$. As we discussed, LPM elicitation problem, where queries are asked on a sphere S_λ has a dual form, where we use a $(q-1)$ dimensional bijective parametrization based on $\boldsymbol{\theta}$ to denote the points on the surface of the sphere. Notice that this parametrization is a function of sin and cos and hence it is Lipschitz as well. Due to monotonicity condition, we assume that the points lie on one orthant of the sphere. Now, suppose the true oracle's metric is denoted by $\mathbf{a}^*$, where $a_i^* = \Pi_{j=1}^{i-1}\sin\theta_j\cos\theta_i$ for $i \in [q-1]$ and $a_q^* = \Pi_{j=1}^{q-1}\sin\theta_j$. Let us denote this parametrization of LPMs by Υ, i.e. $\mathbf{a}^* = \Upsilon(\boldsymbol{\theta}^*)$. This hyperplane is tangent to the unit sphere on a particular point whose coordinates are $\Upsilon(\boldsymbol{\theta}^*)$ itself. Since the metric is linear, by posing pairwise comparisons to the oracle, we ask which off-diagonal confusion is closer to the hyperplane. So, to reach the tangent point on the boundary of the sphere by pairwise comparisons, we are actually decreasing a distance-like function $f^*(\mathbf{c})$ shown in Figure B.3 (right). This function can be represented as $f^*(\boldsymbol{\theta}) = 1 - \langle\Upsilon(\boldsymbol{\theta}^*), \Upsilon(\boldsymbol{\theta})\rangle$ where $\Upsilon(\boldsymbol{\theta}^*)$ are fixed coefficients and $\boldsymbol{\theta}$ changes in our algorithm. This is equivalent to the f function discussed above. Thus using the above guarantees, after $z_1\log(z_2/(q\epsilon^2))(q-1)\log(1/\epsilon)$ queries to the oracle, where z_1, z_2 are constants independent on ϵ and q, we have:

$$\begin{aligned} f^*(\boldsymbol{\theta}) - f^*(\boldsymbol{\theta}^*) &= f^*(\boldsymbol{\theta}) - 0 \\ &= 1 - \langle\Upsilon(\boldsymbol{\theta}^*), \Upsilon(\boldsymbol{\theta}))\rangle \\ &\leq z_3 q\epsilon^2, \end{aligned} \tag{B.13}$$

where z_3 is a constant depending on curvature of the above function f. This implies that:

$$\begin{aligned}
\|\mathbf{a}^* - \widehat{\mathbf{a}}\|_2^2 &= \|\mathbf{a}^*\|_2^2 + \|\widehat{\mathbf{a}}\|_2^2 - 2\langle \mathbf{a}^*, \widehat{\mathbf{a}}\rangle \\
&= 2(1 - \langle \mathbf{a}^*, \widehat{\mathbf{a}}\rangle) \\
&\leq 2z_3 q\epsilon^2. \qquad \text{(B.14)}
\end{aligned}$$

Using the inequality proved before we have that $\|\mathbf{a}^* - \widehat{\mathbf{a}}\|_2 \leq O(\sqrt{q}\epsilon)$. Therefore, in $O\left(T \log \frac{1}{\epsilon}\right)$, we can achieve a point $O(\sqrt{q}\epsilon)$ close to the minimizer, where the number of iterations $T \geq z_1 \log(z_2/(q\epsilon^2))(q-1)$. The term $z_1 \log(z_2/(q\epsilon^2))$ can be considered as the number of cycles, but due to the curvature of the sphere, we find that it is not a dominating factor in the query complexity. For example, when working with a sphere and $\epsilon = 10^{-2}$, two cycles (i.e. $T = 2(q-1)$ in Algorithm 4.2) suffices in practice. Thus, updating each θ_j twice in cycles is sufficient for obtaining the required metric.

It remains to show that, whenever the queried angle is at least $\sqrt{3\epsilon_\Omega/\lambda}$ from the optimal angle, then the oracle gives a correct response. To see this, restrict attention to the hyperplane in which the current angle is moving, say j, for the binary-search phase of the loop. Let θ_j^* be the optimal angle. Observe that for any θ_j such that $\lambda\cos(\theta_j - \theta_j^*) \geq \lambda - \epsilon_\Omega$, the oracle may return a false value. This is because the performance metric is a 1-Lipschitz linear map, and the optimal value on the sphere of radius λ is λ. However, $\cos(x) \leq 1 - x^2/3$, and so for $|\theta_j - \theta_j^*| \geq \sqrt{3\epsilon_\Omega/\lambda}$, we have $\lambda\cos(\theta_j - \theta_j^*) \leq \lambda - \lambda(3\epsilon_\Omega/\lambda)/3 = \lambda - \epsilon_\Omega$. Therefore, so long as $|\theta_j - \theta_j^*| \geq \sqrt{3\epsilon_\Omega/\lambda}$, the oracle provides a correct answer, and the binary search proceeds in the correct direction. QED.

B.4.1 Finding the Sphere $\mathcal{S}_\lambda$

Now, we discuss how a sufficiently large sphere $\mathcal{S}_\lambda$ with radius λ may be found. Consider the following optimization problem, which is a special case of OP2 in [15]. This problem corresponds to feasiblity check problem for a given off-diagonal confusion $\mathbf{c}^0$ for small $\delta \in \mathbb{R}$.

$$\min_{\mathbf{c}\in\mathcal{C}} 0 \qquad s.t. \;\; \|\mathbf{c} - \mathbf{c}^0\|_2 \leq \delta \qquad \text{(B.15)}$$

If a solution to the above problem exists, then Algorithm 1 of [15] returns it. Basically, the approach in [15] will try to construct a classifier whose off-diagonal confusions are δ-close to the given off-diagonal confusion $\mathbf{c}^0$. Hence, checking the feasibility.

Algorithm B.1 computes a value of $\lambda \geq \widetilde{r}/k$, where $\widetilde{r}$ is the radius of the largest ball contained in the set $\mathcal{C}$. Notice that this algorithm is run offline and does not impact query

Algorithm B.1 Approximating the λ Radius

```
Input: The center $o$ of the feasible region of classifiers.
for $j = 1, 2, \cdots, q$ do
    Let $\mathbf{e}_j$ be the standard basis vector for the $j$-th dimension.
    Compute the maximum $\ell_j$ such that $o + \ell_j \mathbf{e}_j$ is feasible by solving (B.15).
end for
Let $CONV$ be the convex hull of $\{o \pm \ell_j \mathbf{e}_j\}_{j=1}^{q}$.
Compute the radius $r$ of the largest ball which can fit inside of $CONV$, centered at $o$.
Output: $\lambda = r$.
```

complexity. Notice that the approach in [15] is consistent, thus we should get a good estimate of the sphere, provided we have sufficient samples.

Lemma B.2. Let $\widetilde{r}$ be the radius of the largest ball centered at o which fits in the feasible space of classifiers. Then Algorithm B.1 returns a radius $\lambda \geq \widetilde{r}/k$.

Proof. Let ℓ_j be as computed in the algorithm, and let $\ell := \min_j \ell_j$. We must have $\ell \geq \widetilde{r}$. Furthermore, the region $CONV$ contains the convex hull of $\{o \pm \ell \mathbf{e}_j\}_{j=1}^{q}$. But this region contains a ball of radius $\ell/\sqrt{q} = \ell/\sqrt{k^2 - k} \geq \ell/k \geq \widetilde{r}/k$, and so $\lambda \geq \widetilde{r}/k$. QED.

B.5 PROOFS OF SECTION 4.4

Proof of Proposition 4.4. We can add a large positive constant if for any $\mathbf{d} \in \mathcal{D}$, $\psi(\mathbf{d}) < 0$. The metric would remain linear fractional. So, it is sufficient to assume $\psi(\mathbf{d}) \geq 0$. Furthermore, boundedness and scale invariance of ψ implies $\psi(\mathbf{d}) \in [0, 1]$, without compromising the linear-fractional form. Now, we look at the sufficient conditions for monotonicity in $\{d_i\}_{i=1}^{k}$ and the numerator and denominator to be positive. Consider the derivative:

$$\frac{\partial \psi}{\partial d_1} = \frac{a_1}{\sum_i b_i d_i + b_0} - \frac{b_1 (\sum_i a_i d_i)}{(\sum_i b_i d_i + b_0)^2} \geq 0 \tag{B.16}$$

Assuming denominator is positive, we have the numerator to be positive and

$$a_1 \geq b_1 \frac{\sum_i a_i d_i}{\sum_i b_i d_i + b_0} \implies a_1 \geq b_1 \sup_{\mathbf{d} \in \mathcal{D}} \frac{\sum_i a_i d_i}{\sum_i b_i d_i + b_0} \implies a_i \geq b_i \overline{\tau} \tag{B.17}$$

The above condition is necessary. Since $\overline{\tau} \in [0, 1]$, by considering all the three cases $b_i = 0, b_i > 0, b_i < 0$, the following are the sufficient conditions for monotonicity: $a_1 \geq b_1$ and $a_1 \geq 0$. Similarly, this is true for all a_i's and b_i's i.e. $a_i \geq b_i, a_i \geq 0 \; \forall \; i \in [k]$ for monotonically increasing DLFPMs. Furthermore, as we assumed that $\psi \in [0,1]$ i.e.

$$\frac{\sum_i a_i d_i}{\sum_i b_i d_i + b_0} \leq 1 \implies \sum_i (a_i - b_i) d_i \leq b_0 \tag{B.18}$$

So, it is sufficient to take $b_0 = \sum_i (a_i - b_i)\zeta_i$ to make the metric bounded in $[0,1]$ and denominator positive. In addition, we can divide the numerator and denominator by $\sum_i a_i$ without changing the metric ψ. Therefore, we take $\sum_i a_i = 1$ during the elicitation task. QED.

Proof of Proposition 4.5. We continue from Equation (4.12), where we saw that $\alpha \geq 0$. Additionally, we ignore the case when $\alpha = 0$, since this would imply a constant ψ^*. Next, we may divide the above equations by $\alpha > 0$ on both sides so that all the coefficients $\mathbf{a}^*$ and $\mathbf{a}^*$ are factored by α. This does not change the metric ψ^*; thus, the SoE becomes:

$$a'_i - \overline{\tau}^* b'_i = \bar{s}_i \;\; \forall\, i \in [k], \quad \overline{\tau}^* b'_0 = \langle \bar{\mathbf{s}}, \overline{\mathbf{d}}^* \rangle. \tag{B.19}$$

Notice that none of the conditions in Assumption 4.3 are changed except $\sum_i a_i = 1$. However, we may still use this condition to learn a constant α times the true metric, which does not harm the elicitation problem. From the last equation, we have that $\overline{\tau} = \langle \bar{\mathbf{s}}, \overline{\mathbf{d}}^* \rangle / b'_0$. Putting this into rest of the equations gives us:

$$\frac{a'_i - \bar{s}_i}{\langle \bar{\mathbf{s}}, \overline{\mathbf{d}}^* \rangle} = \frac{b'_i}{b'_0}. \tag{B.20}$$

By replacing b'_i in the rest of equations further gives us the solution mentioned in the proposition. QED.

Proof of Proposition 4.6. Recall that our metric ϕ is monotonically decreasing in c_i's. As LFPMs are transitional and scale invariant, w.l.o.g., we can assume that $\phi \in [-1,0]$. Taking the derivative in c_1 gives us:

$$\frac{\partial \phi}{\partial c_1} = \frac{a_1}{\sum_i b_i a_i + b_0} - \frac{b_1 (\sum_i a_i c_i)}{(\sum_i b_i c_i + b_0)^2} \leq 0. \tag{B.21}$$

Assuming denominator is positive, we have the numerator to be negative and

$$a_1 \leq b_1 \frac{\sum_i a_i c_i}{\sum_i b_i c_i + b_0} \implies \leq b_1 \phi(\mathbf{c}) \implies b_1 . \underline{\tau} \tag{B.22}$$

The above condition is necessary. Since $\underline{\tau} \in [-1,0]$, by considering all the cases i.e. $b_i = 0, b_i > 0, b_i < 0$ the following are the sufficient condition for monotonicity decreasing

LFPMs: $a_1 \leq -b_1$ and $a_1 \leq 0$. Similarly, this is true for $a_i \leq -b_i, a_i \leq 0 \;\forall\; i \in [q]$ for monotonically decreasing LFPMs. Furthermore, as we assumed that $\phi \in [-1, 0]$, i.e.,

$$\frac{\sum_i a_i c_i}{\sum_i b_i c_i + b_0} \geq -1 \implies \sum_i -(a_i + b_i)c_i \leq b_0 \tag{B.23}$$

Again, so it is sufficient to take $b_0 = \sum_i -(a_i + b_i)\zeta_i$ to make the metric bounded in $[-1, 0]$ and denominator positive. In addition, we can divide the numerator and denominator by $\sum_i |a_i|$ without changing the metric ϕ. This gives us the condition $\sum_i a_i = -1$. QED.

Proof of Proposition 4.7. We start from (4.19), where we saw $\alpha \geq 0$. Additionally, we ignore the case when $\alpha = 0$, since this would imply a constant ϕ^*. Next, we may divide the above equations by $\alpha > 0$ on both sides so that all the coefficients a_i^*'s and b_i^*'s are factored by α. This does not change ϕ^*; thus, the SoE becomes:

$$a_i' - \overline{\tau}^* b_i' = \overline{s}_i, \;\;\forall\;\; i \in [q], \qquad \overline{\tau}^* b_0' = \langle \overline{\mathbf{s}}, \overline{\mathbf{c}}^* \rangle. \tag{B.24}$$

Notice that none of the conditions in Assumption 4.4 are changed except $\sum_i a_i = -1$. However, we may still use this condition to learn a constant α times the true metric, which does not harm the elicitation problem. Similar to DLFPMs, if we somehow know the true a_i''s, we can elicit the LFPM upto a constant multiple. From the last equation, we have that $\overline{\tau} = \langle \overline{\mathbf{s}}, \overline{\mathbf{c}}^* \rangle / b_0'$. Putting this into rest of the equations gives us:

$$\frac{a_i' - \overline{s}_i}{\langle \overline{\mathbf{s}}, \overline{\mathbf{c}}^* \rangle} = \frac{b_i'}{b_0}. \tag{B.25}$$

By replacing b_i in the rest of equations gives us the solution mentioned in the proposition. QED.

B.6 EXTENDED EXPERIMENTS

In this section, we empirically validate the theory and investigate the sensitivity and robustness due to finite sample estimates. For the ease of judgments, we show results corresponding to classes $k = 3$ and $k = 4$. The results and discussion extends to larger number of classes as well. To show the efficacy of the proposed methods, we run experiments on standard machine learning datasets.[1]

[1] The datasets can be downloaded from: https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/multiclass.html, www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/multiclass.html

Table B.1: DLPM elicitation at $\epsilon = 0.01$ for synthetic data. The number of queries used for $k = 3$ and $k = 4$ is 56 and 84, respectively. Since the digits are rounded to two decimal places, $\|\mathbf{a}^*\|_1$ or $\|\widehat{\mathbf{a}}\|_1$ might not be exactly equal to one.

Classes $k = 3$		Classes $k = 4$	
$\psi^* = \mathbf{a}^*$	$\widehat{\psi} = \widehat{\mathbf{a}}$	$\psi^* = \mathbf{a}^*$	$\widehat{\psi} = \widehat{\mathbf{a}}$
(0.21, 0.59, 0.20)	(0.21, 0.60, 0.20)	(0.13, 0.37, 0.12, 0.38)	(0.13, 0.37, 0.12, 0.38)
(0.44, 0.26, 0.31)	(0.44, 0.26, 0.31)	(0.21, 0.26, 0.31, 0.22)	(0.21, 0.26, 0.31, 0.22)
(0.46, 0.33, 0.22)	(0.46, 0.33, 0.22)	(0.23, 0.17, 0.11, 0.48)	(0.23, 0.17, 0.11, 0.48)
(0.23, 0.15, 0.62)	(0.23, 0.15, 0.62)	(0.25, 0.13, 0.45, 0.18)	(0.25, 0.12, 0.45, 0.18)
(0.31, 0.15, 0.54)	(0.3, 0.15, 0.54)	(0.22, 0.17, 0.31, 0.29)	(0.22, 0.17, 0.31, 0.29)
(0.29, 0.40, 0.31)	(0.29, 0.40, 0.31)	(0.38, 0.21, 0.22, 0.20)	(0.38, 0.21, 0.21, 0.20)
(0.35, 0.32, 0.33)	(0.35, 0.33, 0.33)	(0.22, 0.13, 0.14, 0.52)	(0.22, 0.13, 0.14, 0.52)
(0.33, 0.35, 0.32)	(0.33, 0.35, 0.31)	(0.58, 0.17, 0.08, 0.18)	(0.58, 0.17, 0.08, 0.18)

B.6.1 DLPM and LPM Elicitation on Simulated Data (Extended)

We show an extended set of results for the experimental setting discussed in Section 4.6.1. Table B.1 and Table B.2 show elicitation results on the simulated data for DLPMs and LPMs, respectively. We verify that our algorithms elicit the true metrics even for $\epsilon = 0.01$, and as expected, require $4(k-1)\lceil\log(1/\epsilon)\rceil$ and $4T\lceil\log(\pi/2\epsilon)\rceil$ queries for DLPM and LPM elicitation, respectively, where $\lceil\cdot\rceil$ is the ceil function and $T = 2(q-1)$.

B.6.2 Effect of Sphere Size on LPM Elicitation

For real-world datasets, Algorithm 4.2 is agnostic to the error from $\widehat{\eta}_i$'s as long as we get a sphere inside the feasible region of sufficient size. With the following experiment, we show that we incur errors in elicitation when the radius λ is of the order of ϵ_Ω. Recall that, when we are working in a simulated setting, a good proxy for ϵ_Ω is the practical computation error.

Here, we work with $k = 4$ classes. We took $\lambda = 2.500 \times 10^{-12}$ and performed elicitation by considering three spheres of size $^1/_2\lambda$, $^3/_4\lambda$, and λ. We randomly selected hundered DLPMs i.e. $\mathbf{a}^*$'s. We then used Algorithm 4.2 with $\epsilon = 0.01$ to recover the estimates $\widehat{\mathbf{a}}$'s. In Table B.3, we report the proportion of the number of times $\|\mathbf{a}^* - \widehat{\mathbf{a}}\|_\infty \leq \omega$ for different values of ω. We see improved elicitation when we work with λ and incur more errors when the sphere's radius is less than that. In particular, if we take the radius of the order (a little) higher than 10^{-12} then we perform perfect elicitation. Needless to say, when working with real oracle (users), the magnitude of the oracle's feedback noise ϵ_Ω and the size of the sphere will play a role in elicitation performance as suggested in Theorem 4.2.

Table B.2: LPM elicitation at $\epsilon = 0.01$ for synthetic data. The number of queries used for $k = 3$ and $k = 4$ is 320 and 704, respectively. Since the digits are rounded to two decimal places, $\|\mathbf{a}^*\|_2$ or $\|\widehat{\mathbf{a}}\|_2$ might not be exactly equal to one.

Classes	$\phi^* = \mathbf{a}^*$	$\widehat{\phi} = \widehat{\mathbf{a}}$
3	(-0.37, -0.89, -0.09, -0.23, -0.04, -0.03)	(-0.37, -0.89, -0.09, -0.23, -0.04, -0.03)
3	(-0.80, -0.55, -0.18, -0.08, -0.14, -0.05)	(-0.80, -0.55, -0.18, -0.08, -0.14, -0.05)
3	(-0.19, -0.88, -0.28, -0.10, -0.08, -0.30)	(-0.19, -0.88, -0.28, -0.10, -0.08, -0.30)
3	(-0.44, -0.55, -0.33, -0.51, -0.23, -0.28)	(-0.44, -0.55, -0.33, -0.51, -0.23, -0.28)
3	(-0.79, -0.27, -0.25, -0.21, -0.38, -0.23)	(-0.79, -0.27, -0.25, -0.21, -0.38, -0.23)
4	(-0.90, -0.28 -0.10, -0.31, -0.04, -0.05, -0.03, -0.04, -0.02, -0.01, -0.01, -0.01)	(-0.90, -0.28, -0.10, -0.31, -0.04, -0.05, -0.03, -0.04, -0.02, -0.01, -0.01, -0.01)
4	(-0.54, -0.10, -0.62, -0.52, -0.03, -0.07, -0.11, -0.07, -0.14, -0.03, -0.03, -0.04)	(-0.55, -0.11, -0.62, -0.51, -0.03, -0.07, -0.11, -0.07, -0.14, -0.03, -0.03, -0.04)
4	(-0.56, -0.07, -0.79, -0.05, -0.16, -0.16, -0.04, -0.02, -0.03, -0.00, -0.01, -0.01)	(-0.56, -0.07, -0.79, -0.05, -0.16, -0.17, -0.04, -0.02, -0.03, -0.00, -0.01, -0.01)
4	(-0.60, -0.79, -0.09, -0.01, -0.01, -0.02, -0.02, -0.01, -0.01, -0.01, -0.00, -0.00)	(-0.60, -0.79, -0.09, -0.01, -0.01, -0.02, -0.02, -0.01, -0.01, -0.01, -0.00, -0.00)
4	(-0.45, -0.38, -0.42, -0.19, -0.21, -0.63, -0.09, -0.00, -0.00, -0.00, -0.01, -0.01)	(-0.46, -0.38, -0.41, -0.19, -0.20, -0.62, -0.09, -0.00, -0.00, -0.00, -0.01, -0.01)

B.6.3 DLFPM and LFPM Elicitation

Now, we validate elicitation for DLFPMs for classes $k = 3$ and $k = 4$ using the routine discussed in Section 4.4.1. We use the same distribution setting of Section 4.6.1 for both the classes. We define a true metric ψ^* by $\{\mathbf{a}^*, \mathbf{b}^*, b_0^*\}$. Then, we run Algorithm 4.1 with $\epsilon = 0.01$ to find the hyperplane $\bar{\ell}$ and maximizer on $\partial\mathcal{D}^+$, Algorithm 4.3 with $\epsilon = 0.01$ to find the hyperplane $\underline{\ell}$ and minimizer on $\partial\mathcal{D}^-$, and Algorithm 4.4 with $n' = 1000$ (1000 diagonal confusions on $\partial\mathcal{D}^+$ obtained by varying parameter m) and $\delta = 0.01$. This gives us the elicited metric $\widehat{\psi}$, which we represent by $\{\widehat{\mathbf{a}}, \widehat{\mathbf{b}}, \widehat{b}_0\}$. In Table B.4 and Table B.5, we present the elicitation results for DLFPMs for classes $k = 3$ and $k = 4$, respectively. We also present the mean (α) and the standard deviation (σ) of the ratio of the elicited metric $\widehat{\psi}$ to the true metric ψ^* over the set of diagonal confusions used in Algorithm 4.4 (column 3 and 4 of Table B.4 and Table B.5). For a better judgment, we show function evaluations of the true metric and the elicited metric in Figure B.4. The true and the elicited metric are plotted together after vectorizing the set of diagonal confusions in a certain order based on their parametrizations. As expected, we see that the elicited metric is a constant multiple of the true metric.

Now, we validate elicitation for LFPMs for classes $k = 3$ and $k = 4$ using the routine discussed in Section 4.4.2. We define a true metric ϕ^* by $\{\mathbf{a}^*, \mathbf{b}^*, b_0^*\}$. Then, we run Algo-

Table B.3: LPM elicitation on sphere with varying radius and $\epsilon = 0.01$. For randomly chosen hundred $\mathbf{a}^*$, we show the fraction of times our estimates $\widehat{\mathbf{a}}$ obtained with $4 \times 2(q-1)\lceil \log(1/\epsilon) \rceil$ queries satisfy $\|\mathbf{a}^* - \widehat{\mathbf{a}}\|_\infty \leq \omega$. Notice that we incur error only when the radius is of the order of practical computation error, which can be attributed to ϵ_Ω in the simulated setting.

λ \ ω	0.02	0.04	0.06	0.08	0.10
1.250×10^{-12}	0.03	0.38	0.74	0.92	0.94
1.875×10^{-12}	0.09	0.49	0.77	0.94	0.98
2.500×10^{-12}	0.12	0.73	0.93	0.97	0.99

Table B.4: DLFPM Elicitation for synthetic distribution for $k = 3$ classes with $\epsilon = 0.01$. $(\mathbf{a}^*, \mathbf{b}^*, b_0^*)$ denote the true DLFPM. $(\widehat{\mathbf{a}}, \widehat{\mathbf{b}}, \widehat{b}_0)$ denote the elicited LFPM. We empirically verify that the elicited metric is constant multiple (α) of the true metric.

True Metric	Results on Synthetic Distribution (Appendix B.6.3)		
(a_1^*, a_2^*, a_3^*), $(b_1^*, b_2^*, b_3^*), b_0^*$	$(\widehat{a}_1, \widehat{a}_2, \widehat{a}_3)$, $(\widehat{b}_1, \widehat{b}_2, \widehat{b}_3), \widehat{b}_0$	α	σ
(0.21, 0.59, 0.20), (0.11, -0.22, -0.27), 0.41	(0.25, 0.58, 0.18), (0.20, -0.03, -0.17), 0.29	1.23	0.03
(0.45, 0.27, 0.29), (0.39, 0.22, -0.76), 0.43	(0.46, 0.34, 0.20), (0.42, 0.30, -0.73), 0.38	1.03	0.04
(0.08, 0.42, 0.50), (0.07, -0.63, 0.20), 0.37	(0.16, 0.38, 0.47), (0.17, -0.41, 0.23), 0.27	1.22	0.05

rithm 4.2 with $\epsilon = 0.01$ to find the hyperplane $\overline{\ell}$ and maximizer on $\partial\mathcal{S}_\lambda^-$, Algorithm 4.5 with $\epsilon = 0.01$ to find the hyperplane $\underline{\ell}$ and minimizer on $\partial\mathcal{S}_\lambda^+$, and Algorithm 4.6 with $n' = 1000$ (1000 off-diagonal confusions on $\partial\mathcal{S}_\lambda^-$ obtained by varying parameter $\boldsymbol{\theta}$) and $\delta = 0.01$. This gives us the elicited metric $\widehat{\phi}$, which we represent by $\{\widehat{\mathbf{a}}, \widehat{\mathbf{b}}, \widehat{b}_0\}$. In Table B.6, we present the elicitation results for LFPMs for classes $k = 3$. We also present the mean (α) and the standard deviation (σ) of the ratio of the elicited metric $\widehat{\phi}$ to the true metric ϕ^* over the set of off-diagonal confusions used in Algorithm 4.6 (column 3 and 4 of Table B.6).

For a better judgment, we show function evaluations of the true metric and the elicited metric evaluated on selected off-diagonal confusions in the top row of Figure B.5. Due to many terms in the LFPM for $k = 4$, we skip providing true metric and the elicited metric and only mention the α and σ of the true and elicited metric similar to Table B.6. We obtained $\alpha = 0.79, 0.72, 0.72$ and $\sigma = 0.007, 0.007, 0.006$ for the three metrics plotted in the bottom

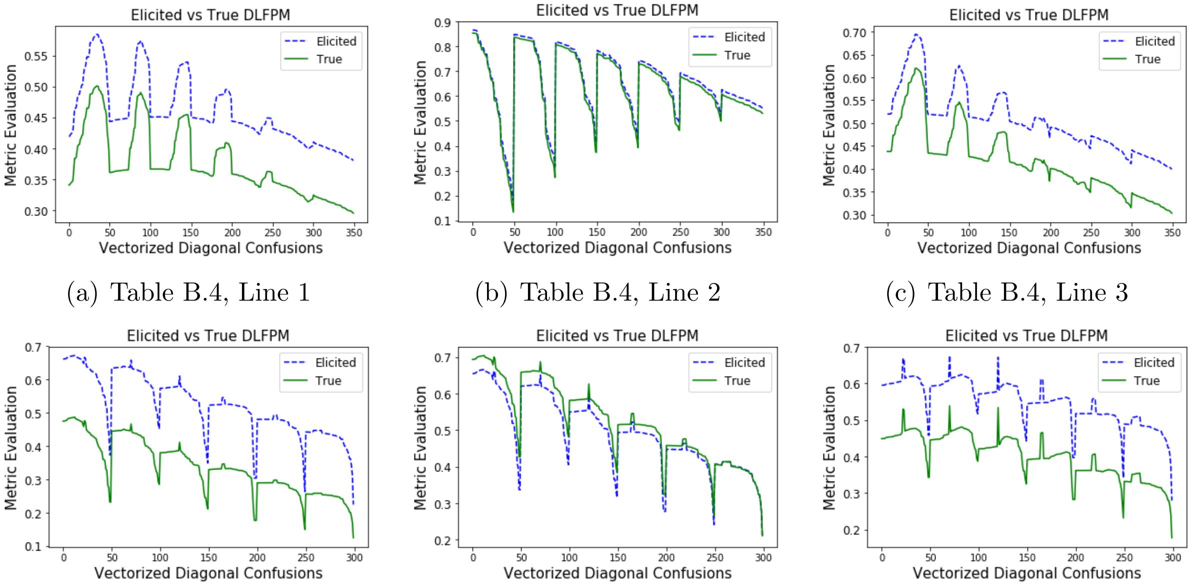
Table B.5: DLFPM Elicitation for synthetic distribution for $k = 4$ classes with $\epsilon = 0.01$. $(\mathbf{a}^*, \mathbf{b}^*, b_0^*)$ denote the true DLFPM. $(\widehat{\mathbf{a}}, \widehat{\mathbf{b}}, \widehat{b}_0)$ denote the elicited LFPM. We empirically verify that the elicited metric is constant multiple (α) of the true metric.

True Metric	Results on Synthetic Distribution (Appendix B.6.3)		
$(a_1^*, a_2^*, a_3^*, a_4^*)$, $(b_1^*, b_2^*, b_3^*, b_4^*), b_0^*$	$(\widehat{a}_1, \widehat{a}_2, \widehat{a}_3, \widehat{a}_4)$, $(\widehat{b}_1, \widehat{b}_2, \widehat{b}_3, \widehat{b}_4), \widehat{b}_0$	α	σ
(0.32, 0.35, 0.06, 0.27), (-1, -0.3, -0.32, 0.25), 0.6	(0.2, 0.29, 0.19, 0.32), (-0.4, -0.01, 0.08, 0.33), 0.26	1.58	0.12
(0.31, 0.22, 0.27, 0.2), (-0.17, -0.01, 0.18, 0.09), 0.25	(0.2, 0.3, 0.26, 0.24), (-0.38, 0.07, 0.16, 0.14), 0.28	0.95	0.04
(0.22, 0.16, 0.41, 0.21), (-0.22, -0.43, -0.18, 0.14), 0.33	(0.19, 0.2, 0.35, 0.26), (-0.09, -0.12, -0.03, 0.24), 0.19	1.38	0.06

row of Figure B.5. The true and the elicited metric are plotted together after vectorizing the set of confusions in a certain order based on their parametrizations. As expected, the elicited metric is a constant multiple of the true metric for both $k = 3$ and $k = 4$.

Table B.6: LFPM Elicitation for $k = 3$ classes with $\epsilon = 0.01$. $(\mathbf{a}^*, \mathbf{b}^*, b_0^*)$ denote the true LFPM. There are thirteen terms to elicit in LFPM. $(\widehat{\mathbf{a}}, \widehat{\mathbf{b}}, \widehat{b}_0)$ denote the elicited LFPM. We empirically verify that the elicited metric is constant multiple (α) of the true metric.

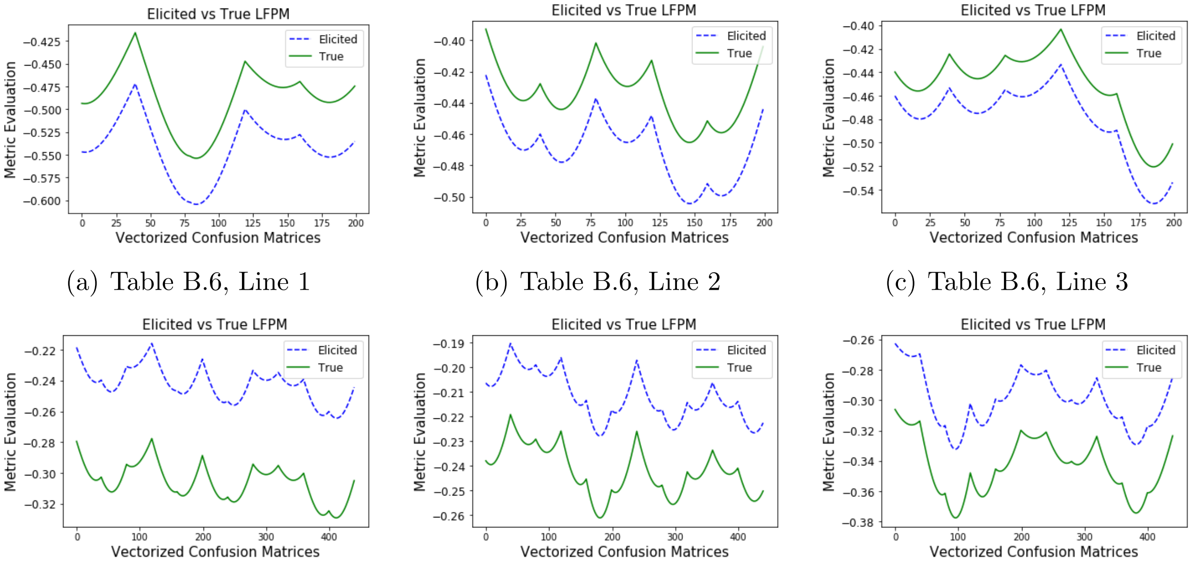
True Metric	Results on Synthetic Distribution (Appendix B.6.3)		
$(a_1^*, a_2^*, a_3^*, a_4^*, a_5^*, a_6^*)$, $(b_1^*, b_2^*, b_3^*, b_4^*, b_5^*, b_6^*), b_0^*$	$(\widehat{a}_1, \widehat{a}_2, \widehat{a}_3, \widehat{a}_4, \widehat{a}_5, \widehat{a}_6)$, $(\widehat{b}_1, \widehat{b}_2, \widehat{b}_3, \widehat{b}_4, \widehat{b}_5, \widehat{b}_6), \widehat{b}_0$	α	σ
(-0.16, -0.05, -0.29, -0.21, -0.17, -0.12), (-0.76, 0.02, -0.88, 0.09, -0.23, -0.38), 2.36	(-0.11, -0.08, -0.15, -0.17, -0.24, -0.25), (-0.66, 0.07, -0.86, 0.04, -0.04, -0.09), 1.89	1.11	0.01
(-0.17, -0.19, -0.09, -0.18, -0.16, -0.2), (-0.3, -0.74, -0.54, -0.37, -0.89, -0.14), 2.99	(-0.05, -0.08, -0.11, -0.16, -0.31, -0.31), (-0.46, -0.82, -0.43, -0.34, -0.48, 0.09), 2.58	1.08	0.01
(-0.3, -0.08, -0.1, -0.12, -0.21, -0.18), (-0.24, -0.52, -0.45, 0, -0.41, -0.94), 2.67	(-0.06, -0.08, -0.11, -0.15, -0.27, -0.33), (-0.59, -0.45, -0.37, 0.07, -0.24, -0.57), 2.36	1.07	0.01



(a) Table B.4, Line 1 (b) Table B.4, Line 2 (c) Table B.4, Line 3

(d) Table B.5, Line 1 (e) Table B.5, Line 2 (f) Table B.5, Line 3

Figure B.4: True and elicited DLFPMs for synthetic distribution from Tables B.4 and B.5. The solid green curve and the dashed blue curve are the true and the elicited metric, respectively. We see that the elicited DLFPMs are constant multiple of the true metrics.



(a) Table B.6, Line 1 (b) Table B.6, Line 2 (c) Table B.6, Line 3

(d) LFP Metric 1, $k = 4$ (e) LFP Metric 2, $k = 4$ (f) LFP Metric 3, $k = 4$

Figure B.5: True and elicited LFPMs. The plots in the top row correspond to the metrics in Table B.6 for $k = 3$. The bottom row corresponds to metrics for $k = 4$. The solid green curve and the dashed blue curve are the true and the elicited metric, respectively. We see that the elicited LFPMs are constant multiple of the true metrics.

APPENDIX C: FAIR PERFORMANCE METRIC ELICITATION

C.1 PROOFS AND DETAILS OF SECTION 5.2

Proof of Proposition 5.1. The set of rates $\mathcal{R}^g$ for a group g satisfies the following properties:

- *Convex*: Let us take two classifiers $h_1^g, h_2^g \in \mathcal{H}^g$ which achieve the rates $\mathbf{r}_1^g, \mathbf{r}_2^g \in \mathcal{R}^g$. We need to check whether or not the convex combination $\alpha \mathbf{r}_1^g + (1-\alpha)\mathbf{r}_2^g$ is feasible, i.e., there exists some classifier which achieve this rate. Consider a classifier h^g, which with probability α predicts what classifier h_1^g predicts and with probability $1-\alpha$ predicts what classifier h_2^g predicts. Then the elements of the rate matrix $R_{ij}^g(h)$ is given by:

$$\begin{aligned} R_{ij}^g(h) &= \mathbb{P}(h^g = j | Y = i) \\ &= \mathbb{P}(h_1^g = j | h^g = h_1^g, Y = i)\mathbb{P}(h^g = h_1^g) + \mathbb{P}(h_2^g = j | h^g = h_2^g, Y = i)\mathbb{P}(h^g = h_2^g) \\ &= \alpha \mathbf{r}_1^g + (1-\alpha)\mathbf{r}_2^g. \end{aligned} \tag{C.1}$$

  Therefore, $\mathcal{R}^g \; \forall \; g \in [m]$ is convex.

- *Bounded:* Since $R_{ij}^g(h) = P[h = j | Y = i] = P[h = j, Y = i]/P[Y = i] \leq 1$ for all $i, j \in [k]$, $\mathcal{R}^g \subseteq [0,1]^q$.

- $\mathbf{e}_i$*'s and* $\mathbf{o}$ *are always achieved:* The classifier which always predicts class i, will achieve the rate $\mathbf{e}_i$. Thus, $\mathbf{e}_i \in \mathcal{R}^g \, \forall \, i \in [k], g \in [m]$ are feasible. Just like the convexity proof, a classifier which predicts similar to one of the trivial classifiers with probability $1/k$ will achieve the rates $\mathbf{o}$.

- $\mathbf{e}_i$*'s are vertices:* Any supporting hyperplane with slope $\ell_{1i} < \ell_{1j} < 0$ and $\ell_{1p} = 0$ for $p \in [k], p \neq i, j$ will be supported by $\mathbf{e}_1$ (corresponding to the trivial classifier which predict class 1). Thus, $\mathbf{e}_i$'s are vertices of the convex set. As long as the class-conditional distributions are not identical, i.e., there is some signal for non-trivial classification conditioned on each group (Assumption 5.1), one can construct a ball around the trivial rate $\mathbf{o}$ and thus $\mathbf{o}$ lies in the interior.

QED.

C.1.1 Finding the Sphere $\mathcal{S}_\rho$

In this section, we discuss how a sufficiently large sphere $\mathcal{S}_\rho$ with radius ρ may be found. The following discussion is extended from Chapter 4 (Section B.4.1) to multiple groups setting and provided here for completeness.

Algorithm C.1 Obtaining the sphere $\mathcal{S}_\rho$ with radius ρ

```
Input: The center $\mathbf{o}$ of the feasible region of rates across groups.
for $j = 1, 2, \cdots, q$ do
    Let $\mathbf{r}_j$ be the standard basis vector for the $j$-th dimension.
    Compute the maximum $\ell_j$ such that $\mathbf{o} + \ell_j \mathbf{r}_j$ is feasible for all groups by solving (C.2).
end for
Let $CONV$ be the convex hull of $\{\mathbf{o} \pm \ell_j \mathbf{r}_j\}_{j=1}^q$.
Compute the radius $s$ of the largest ball which can fit inside of $CONV$, centered at $\mathbf{o}$.
Output: Sphere $\mathcal{S}_\rho$ with radius $\rho = s$ centered at $\mathbf{o}$.
```

The following optimization problem is a special case of OP2 in [15, 131]. The problem corresponds to feasiblity check problem for a given rate $\mathbf{r}_0$ achieved by all groups within small error $\epsilon > 0$.

$$\min_{\mathbf{r}^g \in \mathcal{R}^g \, \forall g \in [m]} 0 \qquad s.t. \;\; \|\mathbf{r}^g - \mathbf{r}_0\|_2 \leq \epsilon \quad \forall \, g \in [m]. \tag{C.2}$$

The above problem checks the feasibility and if a solution to the above problem exists, then Algorithm 1 of [15] returns it. The approach in [15] constructs a classifier whose group-wise rates are ϵ-close to the given rate $\mathbf{r}_0$.

Furthermore, Algorithm C.1.1 computes a value of $\rho \geq \widetilde{s}/k$, where $\widetilde{s}$ is the radius of the largest ball contained in the set $\mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$. Notice that the approach in [15] is consistent, thus we should get a good estimate of the sphere, provided we have sufficient samples. The algorithm runs offline and does not impact query complexity.

Lemma C.1. Let $\widetilde{s}$ be the radius of the largest ball centered at $\mathbf{o}$ in $\mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$. Then Algorithm C.1.1 returns a radius $\rho \geq \widetilde{s}/k$.

Proof. Let ℓ_j be as computed in the algorithm and $\ell := \min_j \ell_j$, then we have $\ell \geq \widetilde{s}$. Moreover, the region $CONV$ contains the convex hull of $\{o \pm \ell \mathbf{r}_j\}_{j=1}^q$; however, this region contains a ball of radius $\ell/\sqrt{q} = \ell/\sqrt{k^2 - k} \geq \ell/k \geq \widetilde{s}/k$, and thus $\rho \geq \widetilde{s}/k$. QED.

C.2 DERIVATIONS OF SECTION 5.3

Notice that $\sum_{g=1}^m \boldsymbol{\tau}^g = \mathbf{1}$, i.e., the vector of ones.

C.2.1 Eliciting the Misclassification Cost $\overline{\phi}(\mathbf{r})$; Part 1 in Figure 5.2 and line 2 in Algorithm 5.1

The key to eliciting $\overline{\phi}$ is to remove the effect of fairness violation $\overline{\varphi}$ in the oracle responses. As explained in Section 5.3.1, we run the LPME procedure (Algorithm 4.2) with the q-

dimensional query space $\mathcal{S}_\rho$, binary search tolerance ϵ, the equivalent oracle Ω^{class}. From Remark 5.1, this subroutine returns a slope $\mathbf{f}$ with $\|\mathbf{f}\|_2 = 1$ such that:

$$\frac{(1-\bar{\lambda})a_i}{(1-\bar{\lambda})a_j} = \frac{f_i}{f_j} \implies \frac{a_i}{a_j} = \frac{f_i}{f_j}. \tag{C.3}$$

Thus, we set $\widehat{\mathbf{a}} \coloneqq \mathbf{f}$ (line 2, Algorithm 5.1).

C.2.2 Eliciting the Fairness Violation $\overline{\varphi}(\mathbf{r}^{1:m})$; Part 2 in Figure 5.2 and lines 3-15 in Algorithm 1

Eliciting the Fairness Violation $\overline{\varphi}(\mathbf{r}^{1:m})$ for $m = 2$; lines 3-6 in Algorithm 1: For $m = 2$, we have only one vector of unfairness weights $\mathbf{b}^{12}$, which we now aim to elicit given $\widehat{\mathbf{a}}$. As discussed in Section 5.3.2, we fix trivial rates (through trivial classifiers) to one group and allow non-trivial rates from $\mathcal{S}_\rho$ on another group. This essentially makes the metric in Definition 5.1 linear. The elicitation procedure is as follows.

Fix trivial classifier predicting class 1 for group 2, i.e., fix $h^2(x) = 1 \, \forall \, x \in \mathcal{X}$, and thus $\mathbf{r}^2 = \mathbf{e}_1$. For group 1, we constrain the confusion rates to lie in the sphere $\mathcal{S}_\rho$, i.e., $\mathbf{r}^1 = \mathbf{s}$ for $\mathbf{s} \in \mathcal{S}_\rho$. Then the metric in Definition 5.1 amounts to:

$$\overline{\Psi}((\mathbf{s},\mathbf{e}_1); \bar{\mathbf{a}}, \overline{\mathbf{b}}^{12}, \bar{\lambda}) = (1-\bar{\lambda})\langle \bar{\mathbf{a}} \odot (1-\boldsymbol{\tau}^2), \mathbf{s}\rangle + \bar{\lambda}\langle \overline{\mathbf{b}}^{12}, |\mathbf{e}_1 - \mathbf{s}|\rangle + c_1. \tag{C.4}$$

The above is a function of $\mathbf{s} \in \mathcal{S}_\rho$. Since $\mathbf{e}_i$'s are binary vectors and since $0 \leq \mathbf{s} \leq 1$, the sign of the absolute function with respect to $\mathbf{s}$ can be recovered. Recall that the rates are defined in row major form of the rate matrices, thus $\mathbf{e}_1$ is 1 at every $(k + j * (k-1))$-th coordinate, where $j \in \{0, \ldots, k-2\}$, and 0 otherwise. The coordinates where the confusion rates are 1 in $\mathbf{e}_1$, the absolute function opens with a negative sign (wrt. $\mathbf{s}$) and with a positive sign otherwise. In particular, define a q-dimensional vector $\mathbf{w}_1$ with entries -1 at every $(k + j * (k-1))$-th coordinate, where $j \in \{0, \ldots, k-2\}$, and 1 otherwise. One may then write the metric $\overline{\Psi}$ as:

$$\overline{\Psi}((\mathbf{s},\mathbf{e}_1)\,; \bar{\mathbf{a}}, \overline{\mathbf{b}}^{12}, \bar{\lambda}) = \langle (1-\bar{\lambda})\bar{\mathbf{a}} \odot (\mathbf{1} - \boldsymbol{\tau}^2) + \bar{\lambda}\mathbf{w}_1 \odot \overline{\mathbf{b}}^{12}, \mathbf{s}\rangle + c_1. \tag{C.5}$$

This is again a linear metric elicitation problem where $\mathbf{s} \in \mathcal{S}$. We may again use the LPME procedure (Algorithm 4.2), which outputs a (normalized) slope $\breve{\mathbf{f}}$ with $\|\breve{\mathbf{f}}\|_2 = 1$ in line 4 of Algorithm 5.1. Using Remark 5.1, we get $q-1$ independent equations and may represent every element of $\overline{\mathbf{b}}^{12}$ based on one element, say $\bar{b}^{12}_{k-1}$, i.e.:

$$\frac{\breve{f}_{k-1}}{\breve{f}_i} = \frac{(1-\bar{\lambda})(1-\tau^2_{k-1})\bar{a}_{k-1} + \bar{\lambda}\bar{b}^{12}_{k-1}}{(1-\bar{\lambda})(1-\tau^2_i)\bar{a}_i + \bar{\lambda}w_{1i}\bar{b}^{12}_i} \qquad \forall\, i \in [q].$$

$$\implies \bar{\lambda}\bar{\mathbf{b}}^{12} = \mathbf{w}_1 \odot \left[\left(\frac{(1-\bar{\lambda})(1-\tau^2_{k-1})\bar{a}_{k-1} + \bar{\lambda}\bar{b}^{12}_{k-1}}{\breve{f}_{k-1}}\right)\breve{\mathbf{f}} - (1-\bar{\lambda})((1-\boldsymbol{\tau}^2)\odot\bar{\mathbf{a}})\right]. \quad \text{(C.6)}$$

In order to elicit entire $\bar{\mathbf{b}}^{12}$, we need one more linear relation such as (C.6). So, we now fix the trivial classifier predicting class k for group 2, i.e., fix $h^2(x) = k \,\forall\, \mathbf{x} \in \mathcal{X}$, and thus $\mathbf{r}^2 = \mathbf{e}_k$. For group 1, we constrain the rates to again lie in the sphere $\mathcal{S}_\rho$ i.e. $\mathbf{r}^1 = \mathbf{s}$ for $\mathbf{s} \in \mathcal{S}_\rho$. Since the rate vectors are in row major form of the rate matrices, notice that $\mathbf{e}_k$ is 1 at every $(k-1+j*(k-1))$-th coordinate, where $j \in \{0,\ldots,k-2\}$, and 0 otherwise. In particular, define a q-dimensional vector $\mathbf{w}_k$ with entries -1 at every $(k-1+j*(k-1))$-th coordinate, where $j \in \{0,\ldots,k-2\}$, and 1 otherwise. One may then write the metric $\overline{\Psi}$ as:

$$\overline{\Psi}((\mathbf{s},\mathbf{e}_k);\bar{\mathbf{a}},\bar{\mathbf{b}}^{12},\bar{\lambda}) = (1-\bar{\lambda})\langle\bar{\mathbf{a}}\odot(1-\boldsymbol{\tau}^2),\mathbf{s}\rangle + \bar{\lambda}\langle\bar{\mathbf{b}}^{12},|\mathbf{e}_k-\mathbf{s}|\rangle + c_k. \quad \text{(C.7)}$$

This is a linear metric elicitation problem where $\mathbf{s} \in \mathcal{S}$. Thus, line 5 of Algorithm 5.1 applies LPME subroutine (Algorithm 4.2), which outputs a (normalized) slope $\widetilde{\mathbf{f}}$ with $\|\widetilde{\mathbf{f}}\|_2 = 1$. Using Remark 5.1, we extract the following relation between two of its coordinates, say the $(k-1)$-th and $((k-1)^2+1)$-th coordinates:

$$\frac{\widetilde{f}_{k-1}}{\widetilde{f}_{(k-1)^2+1}} = \frac{(1-\bar{\lambda})(1-\tau^2_{k-1})\bar{a}_{k-1} - \bar{\lambda}\bar{b}^{12}_{k-1}}{(1-\bar{\lambda})(1-\tau^2_{(k-1)^2+1})\bar{a}_{(k-1)^2+1} + \bar{\lambda}\bar{b}^{12}_{(k-1)^2+1}}. \quad \text{(C.8)}$$

Combining equations (C.6) and (C.8) and replacing the true $\bar{\mathbf{a}}$ with the estimated $\widehat{\mathbf{a}}$ from Section 5.3.1, we have an estimate of the scaled substitute as:

$$\widetilde{\mathbf{b}}^{12} = \mathbf{w}_1 \odot \left[\delta\breve{\mathbf{f}}^{12} - \widehat{\mathbf{a}}\odot(\mathbf{1}-\boldsymbol{\tau}^2)\right], \quad \text{(C.9)}$$

$$\text{where } \delta = \frac{2(1-\tau^2_{k-1})\widehat{a}_{k-1}}{\breve{f}_{k-1}}\left[\frac{\frac{(1-\tau^2_{(k-1)^2+1})\widehat{a}_{(k-1)^2+1}}{(1-\tau^2_{k-1})\widehat{a}_{k-1}} - \frac{\widetilde{f}_{(k-1)^2+1}}{\widetilde{f}_{k-1}}}{\left(\frac{\breve{f}_{(k-1)^2+1}}{\breve{f}_{k-1}} - \frac{\widetilde{f}_{(k-1)^2+1}}{\widetilde{f}_{k-1}}\right)}\right],$$

and $\widetilde{\mathbf{b}}$ is a scaled substitute defined as $\widetilde{\mathbf{b}}^{12} := \frac{\bar{\lambda}}{(1-\bar{\lambda})}\bar{\mathbf{b}}^{12}$, which nonetheless is computable from (C.9). Since we require a solution $\widehat{\mathbf{b}}$ such that $\|\widehat{\mathbf{b}}\|_2 = 1$ (Definition 5.1), we normalize $\widetilde{\mathbf{b}}$ and get the final solution:

$$\widehat{\mathbf{b}}^{12} = \frac{\widetilde{\mathbf{b}}^{12}}{\|\widetilde{\mathbf{b}}^{12}\|_2}. \tag{C.10}$$

Notice that, due to normalization, the solution is independent of the true trade-off $\overline{\lambda}$.

Eliciting the Fairness Violation $\overline{\varphi}(\mathbf{r}^{1:m})$ for $m > 2$; line 8-14 in Algorithm 5.1: Consider a non-empty set of sets $\mathcal{M} \subset 2^{[m]} \setminus \{\varnothing, [m]\}$. We will later discuss how to choose $\mathcal{M}$ for efficient elicitation. When $m > 2$, 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 σ. We follow exactly similar procedure as in the previous section, i.e., fixing trivial rates (through trivial classifiers) on the groups in σ and allowing non-trivial rates from $\mathcal{S}_\rho$ on the groups in $[m] \setminus \sigma$. In particular, consider a paramterization $\nu : (\mathcal{S}_\rho, \mathcal{M}, [k]) \to \mathcal{R}^{1:m}$ defined as:

$$\nu(\mathbf{s}, \sigma, i) \coloneqq \mathbf{r}^{1:m} \quad \text{such that} \quad \mathbf{r}^g = \begin{cases} \mathbf{e}_i & \text{if } g \in \sigma \\ \mathbf{s} & \text{o.w.} \end{cases} \tag{C.11}$$

i.e., ν assigns trivial confusion rates $\mathbf{e}_i$ on the groups in σ and assigns $\mathbf{s} \in \mathcal{S}_\rho$ on the rest of the groups. Similar to the previous section, we first fix trivial classifier predicting class 1 for groups in σ and constrain the rates for groups in $[m] \setminus \sigma$ to be on the sphere $\mathcal{S}_\rho$. Such a setup is governed by the parametrization $\nu(\cdot, \sigma, 1)$ in equation (C.11). Specifically, fixing $h^g(\mathbf{x}) = 1 \; \forall \; g \in \sigma$ would entail the metric in Definition 5.1 to be:

$$\overline{\Psi}(\nu(\mathbf{s}, \sigma, 1); \bar{\mathbf{a}}, \overline{\mathbf{B}}, \overline{\lambda}) = (1 - \overline{\lambda})\langle \bar{\mathbf{a}} \odot (\mathbf{1} - \boldsymbol{\tau}^\sigma), \mathbf{s}\rangle + \lambda \langle \overline{\boldsymbol{\eta}}^\sigma, |\mathbf{e}_1 - \mathbf{s}|\rangle + c_1, \tag{C.12}$$

where $\boldsymbol{\tau}^\sigma = \sum_{g \in \sigma} \boldsymbol{\tau}^g$ and $\overline{\boldsymbol{\eta}}^\sigma = \sum_{u,v \in [m], v > u} \mathbf{1}\left[|\{u, v\} \cap \sigma| = 1\right] \overline{\mathbf{b}}^{uv}$. Similar to the previous section, since $\mathbf{e}_i$'s are binary vectors, the sign of the absolute function w.r.t. $\mathbf{s}$ can be recovered. In particular, the metric amounts to:

$$\overline{\Psi}(\nu(\mathbf{s}, \sigma, 1); \bar{\mathbf{a}}, \overline{\mathbf{B}}, \overline{\lambda}) = \langle (1 - \overline{\lambda})\bar{\mathbf{a}} \odot (\mathbf{1} - \boldsymbol{\tau}^2) + \overline{\lambda}\mathbf{w}_1 \odot \overline{\boldsymbol{\eta}}^\sigma, \mathbf{s}\rangle + c_1, \tag{C.13}$$

where $\mathbf{w}_1 \coloneqq 1 - 2\mathbf{e}_1$ and c_1 is a constant not affecting the responses. Notice that (C.12) and (C.13) are analogous to (C.4) and (C.5), respectively, except that $\boldsymbol{\tau}^2$ is replaced by $\boldsymbol{\tau}^\sigma$ and $\overline{\mathbf{b}}^{12}$ is replaced by $\overline{\boldsymbol{\eta}}^\sigma$. This is a linear metric in $\mathbf{s}$. We again the use the LPME procedure in line 10of Algorithm 5.1, which outputs a normalized slope $\breve{\mathbf{f}}^\sigma$ such that $\|\breve{\mathbf{f}}^\sigma\|_2 = 1$, and thus we get an analogous solution to (C.6) as:

$$\overline{\lambda}\overline{\boldsymbol{\eta}}^\sigma = \mathbf{w}_1 \odot \left[\left(\frac{(1 - \overline{\lambda})(1 - \tau^\sigma_{k-1})\bar{a}_{k-1} + \overline{\lambda}\overline{\eta}^\sigma_{k-1}}{\breve{f}^\sigma_{k-1}}\right)\breve{\mathbf{f}}^\sigma - (1 - \overline{\lambda})((\mathbf{1} - \boldsymbol{\tau}^\sigma) \odot \bar{\mathbf{a}}\right]. \tag{C.14}$$

In order to elicit entire $\overline{\boldsymbol{\eta}}^\sigma$, we need one more linear relation such as (C.14). So, we now fix the trivial rates through trivial classifier predicting class k for the groups in σ, i.e., fix $h^g(x) = k \,\forall\, \mathbf{x} \in \mathcal{X}$ if $g \in \sigma$, and thus $\mathbf{r}^g = \mathbf{e}_k$ for all groups $g \in \sigma$. For the rest of the groups, we constrain the confusion rates to again lie in the sphere $\mathcal{S}_\rho$ i.e. $\mathbf{r}^g = \mathbf{s}$ for $\mathbf{s} \in \mathcal{S}_\rho$ for all groups $g \in [m] \setminus \sigma$. Such a setup is governed by the parametrization $\nu(\cdot, \sigma, k)$ (C.11). The metric $\overline{\Psi}$ in Definition 5.1 amounts to:

$$\overline{\Psi}(\nu(\mathbf{s}, \sigma, k); \bar{\mathbf{a}}, \overline{\mathbf{B}}, \bar{\lambda}) = (1 - \bar{\lambda})\langle \bar{\mathbf{a}} \odot (1 - \boldsymbol{\tau}^\sigma), \mathbf{s}\rangle + \bar{\lambda}\langle \overline{\boldsymbol{\eta}}^\sigma, |\mathbf{e}_k - \mathbf{s}|\rangle + c_k. \tag{C.15}$$

Thus by running LPME procedure again in line 11 of Algorithm 5.1 results in $\widetilde{\mathbf{f}}^{12}$ with $\|\widetilde{\mathbf{f}}^{12}\|_2 = 1$. Using Remark 5.1, we extract the following relation between the $(k-1)$-th and $((k-1)^2+1)$-th coordinates:

$$\frac{\widetilde{f}^\sigma_{k-1}}{\widetilde{f}^\sigma_{(k-1)^2+1}} = \frac{(1-\bar{\lambda})(1-\tau^\sigma_{k-1})\bar{a}_{k-1} - \bar{\lambda}\bar{\eta}^\sigma_{k-1}}{(1-\bar{\lambda})(1-\tau^\sigma_{(k-1)^2+1})\bar{a}_{(k-1)^2+1} + \bar{\lambda}\bar{\eta}^\sigma_{(k-1)^2+1}}. \tag{C.16}$$

Combining equations (C.14) and (C.16), we have:

$$\sum\nolimits_{u,v} \mathbf{1}\left[|\{u, v\} \cap \sigma| = 1\right] \widetilde{\mathbf{b}}^{uv} = \boldsymbol{\gamma}^\sigma, \tag{C.17}$$

where

$$\begin{aligned}
\boldsymbol{\gamma}^\sigma &= \mathbf{w}_1 \odot \left[\delta^\sigma \mathbf{f}^\sigma - \widehat{\mathbf{a}} \odot (\mathbf{1} - \boldsymbol{\tau}^\sigma)\right], \\
\delta^\sigma &= \frac{2(1-\tau^\sigma_{k-1})\widehat{a}_{k-1}}{f^\sigma_{k-1}} \left[\frac{\frac{(1-\tau^\sigma_{(k-1)^2+1})\widehat{a}_{(k-1)^2+1}}{(1-\tau^\sigma_{k-1})\widehat{a}_{k-1}} - \frac{\widetilde{f}^\sigma_{(k-1)^2+1}}{\widetilde{f}^\sigma_{k-1}}}{\left(\frac{f^\sigma_{(k-1)^2+1}}{f^\sigma_{k-1}} - \frac{\widetilde{f}^\sigma_{(k-1)^2+1}}{\widetilde{f}^\sigma_{k-1}}\right)} \right],
\end{aligned} \tag{C.18}$$

and $\widetilde{\mathbf{b}}^{uv} := \bar{\lambda}\overline{\mathbf{b}}^{uv}/(1-\bar{\lambda})$ is a scaled version of the true (unknown) $\overline{\mathbf{b}}$, which nonetheless can be computed from (C.17).

By two runs of LPME algorithm, we can get $\boldsymbol{\gamma}^\sigma$ and solve (C.17). However, the left hand side of (C.17) does not allow us to recover the $\widetilde{\mathbf{b}}$'s separately and provides only one equation. Let us denote the Equation (C.17) by ℓ^σ corresponding to the set σ. In order to elicit all $\widetilde{\mathbf{b}}$'s we need a system of $M := \binom{m}{2}$ independent equations in order to elicit the M weight vectors.

This is easily achievable by choosing M σ's so that we get M set of unique equations like (C.17). Let $\mathcal{M}$ be those set of sets.

In most cases, pairing two groups to have trivial rates (through trivial classifiers) and rest of the groups to have rates from the sphere $\mathcal{S}$ will work. For example, when $m = 3$, fixing $\mathcal{M} = \{\{1,2\},\{1,3\},\{2,3\}\}$ suffices. Thus, running over all the choices of sets of groups $\sigma \in \mathcal{M}$ provides the system of equations $\mathcal{L} \coloneqq \cup_{\sigma\in\mathcal{M}} \ell^\sigma$ (line 12 in Algorithm 5.1), which is formally described as follows:

$$\begin{bmatrix} \Xi & 0 & \dots & 0 \\ 0 & \Xi & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \Xi \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{b}}_{(1)} \\ \widetilde{\mathbf{b}}_{(2)} \\ \dots \\ \widetilde{\mathbf{b}}_{(q)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\gamma}_{(1)} \\ \boldsymbol{\gamma}_{(2)} \\ \dots \\ \boldsymbol{\gamma}_{(q)} \end{bmatrix}, \tag{C.19}$$

where $\widetilde{\mathbf{b}}_{(i)} = (\widetilde{b}_i^1, \widetilde{b}_i^2, \cdots, \widetilde{b}_i^M)$ and $\boldsymbol{\gamma}_{(i)} = (\gamma_i^1, \gamma_i^2, \cdots, \gamma_i^M)$ are vectorized versions of the i-th entry across groups for $i \in [q]$, and $\Xi \in \{0,1\}^{M\times M}$ is a binary full-rank matrix denoting membership of groups in the set $\sigma \in \mathcal{M}$. For instance, for the choice of $\mathcal{M} = \{\{1,2\},\{1,3\},\{2,3\}\}$ when $m = 3$ gives:

$$\Xi = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}. \tag{C.20}$$

From technical point of view, one may choose any $\mathcal{M}$ such that the resulting group membership matrix Ξ is non-singular. Hence the solution of the system of equations $\mathcal{L}$ is:

$$\begin{bmatrix} \widetilde{\mathbf{b}}_{(1)} \\ \widetilde{\mathbf{b}}_{(2)} \\ \dots \\ \widetilde{\mathbf{b}}_{(q)} \end{bmatrix} = \begin{bmatrix} \Xi & 0 & \dots & 0 \\ 0 & \Xi & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \Xi \end{bmatrix}^{(-1)} \begin{bmatrix} \boldsymbol{\gamma}_{(1)} \\ \boldsymbol{\gamma}_{(2)} \\ \dots \\ \boldsymbol{\gamma}_{(q)} \end{bmatrix}. \tag{C.21}$$

When we normalize $\widetilde{\mathbf{b}}$, we get the final fairness violation weight estimates as:

$$\widehat{\mathbf{b}}^{uv} = \frac{\widetilde{\mathbf{b}}^{uv}}{\sum_{u,v=1,v>u}^{m} \|\widetilde{\mathbf{b}}^{uv}\|_2} \quad \text{for} \quad u, v \in [m], v > u. \tag{C.22}$$

Notice that, due to the above normalization, the solution is again independent of the true trade-off $\overline{\lambda}$.

C.2.3 Eliciting Trade-off $\overline{\lambda}$; Part 3 in Figure 5.2 and line 16 in Algorithm 5.1

For ease of notation, let us construct a parametrization $\nu' : \mathcal{S}_\varrho^+ \to \mathcal{R}^{1:m}$:

$$\nu'(\mathbf{s}^+) \coloneqq (\mathbf{s}^+, \mathbf{o}, \dots, \mathbf{o}). \tag{C.23}$$

Using the parametrization ν' from (C.23), the metric in Definition 5.1 reduces to a linear metric in $\mathbf{s}^+$ as discussed in (5.22), i.e:

$$\overline{\Psi}(\nu'(\mathbf{s}^+)\,;\,\bar{\mathbf{a}}, \overline{\mathbf{B}}, \bar{\lambda}) = \langle (1-\bar{\lambda})\boldsymbol{\tau}^1 \odot \bar{\mathbf{a}} + \bar{\lambda} \sum\nolimits_{v=2}^{m} \bar{\mathbf{b}}^{1v}, \mathbf{s}^+ \rangle + c. \tag{C.24}$$

We first show the proof of Lemma 5.1 and then discuss the trade-off elicitation algorithm (Algorithm 5.2).

Proof of Lemma 5.1. For simplicity, let us abuse notation for this proof and denote $\boldsymbol{\tau}^1 \odot \bar{\mathbf{a}}$ simply by $\mathbf{a}$, $\sum_{v=2}^{m} \bar{\mathbf{b}}^{1v}$ simply by $\mathbf{b}$, and $\mathcal{S}_\varrho^+$ simply by $\mathcal{S}$.

$\mathcal{S}$ is a convex set. Let $\mathcal{Z} = \{\mathbf{z} = (z_1, z_2) \,|\, z_1 = <\mathbf{a}, \mathbf{s}>, z_2 = <\mathbf{b}, \mathbf{s}>, \mathbf{s} \in \mathcal{S}\}$.

Claim: $\mathcal{Z}$ is convex.

Let $z, z' \in \mathcal{Z}$.

$$\begin{aligned} \alpha z_1 + (1-\alpha)z_1' &= \alpha <\mathbf{a}, \mathbf{s}> + (1-\alpha) <\mathbf{a}, \mathbf{s}'> &= <\mathbf{a}, \alpha\mathbf{s} + (1-\alpha)\mathbf{s}'> \\ \alpha z_2 + (1-\alpha)z_2' &= \alpha <\mathbf{b}, \mathbf{s}> + (1-\alpha) <\mathbf{b}, \mathbf{s}'> &= <\mathbf{b}, \alpha\mathbf{s} + (1-\alpha)\mathbf{s}'> \end{aligned}$$

Since $\alpha\mathbf{s} + (1-\alpha)\mathbf{s}' \in \mathcal{S}$, $\alpha z + (1-\alpha)z' \in \mathcal{Z}$. Hence $\mathcal{Z}$ is convex.

Claim: The boundary of the set $\mathcal{Z}$ is a strictly convex curve with no vertices for $\mathbf{a} \neq \mathbf{b}$.

Recall that, the required function is given by:

$$\vartheta(\lambda) = \max\nolimits_{\mathbf{z}\in\mathcal{Z}} (1-\lambda)z_1 + \lambda z_2 + c \tag{C.25}$$

(i) Since the set $\mathcal{Z}$ is convex, every boundary point is supported by a hyperplane.

(ii) Since $\mathbf{a} \neq \mathbf{b}$, notice that the slope is uniquely defined by λ. Since the sphere $\mathcal{S}$ is strictly convex, the above linear functional defined by λ is maximized by a unique point in $\mathcal{Z}$ (similar to Lemma 4.1). Thus, the the hyperplane is tangent at a unique point on the boundary of $\mathcal{Z}$.

(iii) It only remains to show that there are no vertices on the boundary of $\mathcal{Z}$. Recall that a vertex exists if (and only if) some point is supported by more than one tangent hyperplane in two dimensional space. This means there are two values of λ that achieve the same maximizer. This is contradictory since there are no two linear functionals that achieve the same maximizer on $\mathcal{S}$.

This implies that the boundary of $\mathcal{Z}$ is a strictly convex curve. Since we are interested in the maximization of ϑ, let this boundary be the upper boundary denoted by $\partial\mathcal{Z}_+$.

Claim: Let $v : [0,1] \to \partial\mathcal{Z}_+$ be continuous, bijective, parametrizations of the upper boundary. Let $\vartheta : \mathcal{Z} \to \mathbb{R}$ be a quasiconcave function which is monotone increasing in both

z_1 and z_2. Then the composition $\vartheta \circ \upsilon : [0,1] \to \mathbb{R}$ is strictly quasiconcave (and therefore unimodal with no flat regions) on the interval $[0,1]$.

Let S be some superlevel set of the quasiconcave function ϑ. Since υ is a continuous bijection and since the boundary $\partial\mathcal{Z}_+$ is a strictly convex curve with no vertices, w.l.o.g., for any $r < s < t$, $z_1(\upsilon(r)) < z_1(\upsilon(s)) < z_1(\upsilon(t))$, and $z_2(\upsilon(r)) > z_2(\upsilon(s)) > z_2(\upsilon(t))$. (otherwise, swap r and t). Since the boundary $\partial\mathcal{Z}_+$ is a strictly convex curve, then $\upsilon(s)$ must be greater (component-wise) a point in the convex combination of $\upsilon(r)$ and $\upsilon(t)$. Let us denote that point by u. Since ϑ is monotone increasing, then $x \in S$ implies that $y \in S$, too, for all $y \geq x$ componentwise. Therefore, $\vartheta(\upsilon(s)) \leq \vartheta(u)$. Since S is convex, $u \in S$ and thus $\upsilon(s) \in S$.

This implies that $\upsilon^{-1}(\partial\mathcal{Z}_+ \cap S)$ is an interval; hence it is convex, which in turn tells us that the superlevel sets of $\vartheta \circ \upsilon$ are convex. So, $\vartheta \circ \upsilon$ is quasiconcave, as desired. This implies unimodaltiy, because a function defined on real line which has more than one local maximum can not be quasiconcave. Moreover, since there are no vertices on the boundary $\partial\mathcal{Z}_+$, the $\vartheta \circ \upsilon : [0,1] \to \mathbb{R}$ is strictly quasiconcave (and thus unimodal with no flat regions) on the interval $[0,1]$. This completes the proof of Lemma 5.1. QED.

C.3 PROOF OF SECTION 5.4

Proof of Theorem 5.1. We break this proof into three parts.

1. *Elicitation guarantees for the misclassification cost $\widehat{\phi}$ (i.e., $\widehat{\mathbf{a}}$)*

   Since Algorithm 5.1 elicits a linear metric using the q-dimensional sphere $\mathcal{S}$, the guarantees on $\widehat{\mathbf{a}}$ follows from Theorem 4.2. Thus, under Assumption 5.2, the output $\widehat{\mathbf{a}}$ from line 2 of Algorithm 5.1 satisfies $\|\mathbf{a}^* - \widehat{\mathbf{a}}\|_2 \leq O(\sqrt{q}(\epsilon + \sqrt{\epsilon_\Omega/\rho}))$ after $O\left(q \log \frac{\pi}{2\epsilon}\right)$ queries.

2. *Elicitation guarantees for the fairness violation cost $\widehat{\varphi}$ (i.e., $\widehat{\mathbf{B}}$)*

   We start with the definition of true $\boldsymbol{\gamma}$ (i.e. when all the elicited entities are true) from (C.17) and let us drop the superscript σ for simplicity. Furthermore, let $\epsilon + \sqrt{\epsilon_\Omega/\rho}$ be denoted by ϵ.

$$\boldsymbol{\gamma} = \mathbf{w}_1 \odot \left[\delta\breve{\mathbf{f}} - \bar{\mathbf{a}} \odot (\mathbf{1} - \boldsymbol{\tau})\right], \quad \text{where} \tag{C.26}$$

$$\delta = \frac{2(1-\tau_{k-1})\bar{a}_{k-1}}{\breve{f}_{k-1}} \left[\frac{\frac{(1-\tau_{(k-1)^2+1})\bar{a}_{(k-1)^2+1}}{(1-\tau_{k-1})\bar{a}_{k-1}} - \frac{\widetilde{f}_{(k-1)^2+1}}{\widetilde{f}_{k-1}}}{\left(\frac{\breve{f}_{(k-1)^2+1}}{\breve{f}_{k-1}} - \frac{\widetilde{f}_{(k-1)^2+1}}{\widetilde{f}_{k-1}}\right)} \right]. \tag{C.27}$$

Let us look at the derivative of the i-th coordinate of $\boldsymbol{\gamma}$.

$$\frac{\partial \gamma_i}{\partial a_j} = \begin{cases} 0 & \text{if } j \neq i, j \neq k-1, j \neq (k-1)^2+1 \\ -\tau_i & \text{if } j = i \\ c_{i,1} & \text{if } j = k-1 \\ c_{i,2} & \text{if } j = (k-1)^2+1, \end{cases} \tag{C.28}$$

where $c_{i,1}$ and $c_{i,2}$ are some bounded constants due to Assumption 5.2. Similarly, $\partial\gamma_i/\partial f_j$ is bounded as well due to the regularity Assumption 5.2. This means that γ_i is Lipschitz in ℓ_2-norm w.r.t. $\mathbf{a}$ and $\mathbf{f}$. Thus,

$$\|\boldsymbol{\gamma} - \widehat{\boldsymbol{\gamma}}\|_\infty \leq c_3\|\bar{\mathbf{a}} - \widehat{\mathbf{a}}\|_2 + c_4\|\breve{\mathbf{f}} - \widehat{\breve{\mathbf{f}}}\|_2, \tag{C.29}$$

for some Lipschits constants c_3 and c_4. From the bounds of Part 1 of this proof, we have:

$$\|\boldsymbol{\gamma} - \widehat{\boldsymbol{\gamma}}\|_\infty \leq O(\sqrt{q}\epsilon). \tag{C.30}$$

Recall the construction of $\widetilde{\mathbf{b}}_{(i)}$ from (C.19). We then have from the solution of system of equations (C.21) that:

$$\widetilde{\mathbf{b}}_{(i)} = \Xi^{-1}\boldsymbol{\gamma}_{(i)} \quad \forall\, i \in [q], \tag{C.31}$$

where $\widetilde{\mathbf{b}}_{(i)} = (\widetilde{b}_i^1, \widetilde{b}_i^2, \cdots, \widetilde{b}_i^M)$ and $\widetilde{\boldsymbol{\gamma}}_{(i)} = (\gamma_i^1, \gamma_i^2, \cdots, \gamma_i^M)$ are vectorized versions of the i-th entry across groups for $i \in [q]$. $\Xi \in \{0,1\}^{M\times M}$ is a full-rank symmetric matrix with bounded infinity norm $\|\Xi^{-1}\|_\infty \leq c$ (here, infinity norm of a matrix is defined as the maximum absolute row sum of the matrix). Thus we have: $\|\widetilde{\mathbf{b}}_{(i)} - \widehat{\widetilde{\mathbf{b}}}_{(i)}\|_\infty =$

$$\|\Xi^{-1}\boldsymbol{\gamma}_{(i)} - \Xi^{-1}\widehat{\boldsymbol{\gamma}}_{(i)}\|_\infty = \|\Xi^{-1}(\boldsymbol{\gamma}_{(i)} - \widehat{\boldsymbol{\gamma}}_{(i)})\|_\infty \leq \|\Xi^{-1}\|_\infty\|\boldsymbol{\gamma}_{(i)} - \widehat{\boldsymbol{\gamma}}_{(i)}\|_\infty, \tag{C.32}$$

which gives

$$\|\widetilde{\mathbf{b}}_{(i)} - \widehat{\widetilde{\mathbf{b}}}_{(i)}\|_\infty \leq O(\sqrt{q}\epsilon). \tag{C.33}$$

Now, our final estimate is the normalized form of $\widehat{\widetilde{\mathbf{b}}}$ from (C.22), so the final error in the stacked version $vec(\overline{\mathbf{B}})$ and $vec(\widehat{\mathbf{B}})$ is:

$$\|vec(\overline{\mathbf{B}}) - vec(\widehat{\mathbf{B}})\|_\infty \leq O(\sqrt{q}\epsilon). \tag{C.34}$$

Since there are $q \times M$ entities in $vec(\mathbf{B})$, we have:

$$\|vec(\overline{\mathbf{B}}) - vec(\widehat{\mathbf{B}})\|_2 \leq O(\sqrt{qM}\sqrt{q}\epsilon) = O(mq\epsilon). \tag{C.35}$$

Due to elicitation on sphere and the oracle noise ϵ_Ω as defined in Definition 5.4, we can replace ϵ with $\epsilon + \sqrt{\epsilon_\Omega/\rho}$ back to get the final bound on fairness violation weights as in Theorem 5.1.

3. *Elicitation guarantees for the trade-off parameter (i.e., $\widehat{\lambda}$)*

   The metric for our purpose is a linear metric in $\mathbf{s}^+ \in \mathcal{S}_\rho^+$ with the following slope:

   $$\overline{\Psi}(\nu'''(\mathbf{s}^+)\,;\,\bar{\mathbf{a}}, \overline{\mathbf{B}}, \bar{\lambda}) = \langle (1-\bar{\lambda})\boldsymbol{\tau}^1 \odot \bar{\mathbf{a}} + \bar{\lambda}\sum_{v=2}^{m} \overline{\mathbf{b}}^{1v}, \mathbf{s}^+\rangle. \tag{C.36}$$

   Since we elicit λ through queries over a surface of the sphere, we pose this problem as finding the right angle (slope) defined by the true $\bar{\lambda}$. Note that $\bar{\lambda}$ is what we want to elicit; however, due to oracle noise ϵ_Ω, we can only aim to achieve a target angle λ_t. Moreover, we do not have true $\bar{\mathbf{a}}$ and $\overline{\mathbf{B}}$ but have only estimates $\widehat{\mathbf{a}}$ and $\widehat{\mathbf{B}}$. Thus we query proxy solutions always and can only aim to achieve an estimated version λ_e of the target angle. Lastly, Algorithm 5.2 is stopped within an ϵ threhsold, thus the final solution $\widehat{\lambda}$ is within ϵ distance from λ_e. In total, we want to find:

   $$|\bar{\lambda} - \widehat{\lambda}| \leq \underbrace{|\bar{\lambda} - \lambda_t|}_{\text{oracle error}} + \underbrace{|\lambda_t - \lambda_e|}_{\text{estimation error}} + \underbrace{|\lambda_e - \widehat{\lambda}|}_{\text{optimization error}}. \tag{C.37}$$

   - optimization error: $|\lambda_e - \widehat{\lambda}| \leq \epsilon$.
   - oracle error: Notice that the oracle correctly answers as long as $\varrho(1-\cos(\bar{\lambda}-\lambda_t)) > \epsilon_\Omega$. This is because the metric is a 1-Lipschitz linear function, and the optimal value on the sphere of radius ϱ is ϱ. However, as $1-\cos(x) \geq x^2/3$, so oracle is correct as long as $|\bar{\lambda} - \lambda_e| \geq \sqrt{3\epsilon_\Omega/\varrho}$. Given this, the binary search proceeds in the correct direction.
   - estimation error: We make this error because we only have access to the estimated $\widehat{\mathbf{a}}$ and $\widehat{\mathbf{B}}$ not the true $\bar{\mathbf{a}}$ and $\overline{\mathbf{B}}$. However, since the metric in (C.36) is Lipschitz in $\bar{\mathbf{a}}$ and $\sum_{v=2}^{m} \overline{\mathbf{b}}^{1v}$, this error can be treated as oracle feedback noise where the oracle responses with the estimated $\widehat{\mathbf{a}}$ and $\widehat{\mathbf{B}}$. Thus, if we replace ϵ_Ω from the previous point to the error in $\widehat{\mathbf{a}}$ and $\sum_{v=2}^{m} \widehat{\mathbf{b}}^{1v}$, the binary search moves in the right direction as long as

     $$|\lambda_t - \lambda_e| \geq O\left(\sqrt{\frac{\|\bar{\mathbf{a}} - \widehat{\mathbf{a}}\|_2 + \sum_{v=2}^{m}\|\overline{\mathbf{b}}^{1v} - \widehat{\mathbf{b}}^{1v}\|_2}{\varrho}}\right) = O\left(\sqrt{mq(\epsilon + \sqrt{\epsilon_\Omega/\rho})/\varrho}\right), \tag{C.38}$$

where we have used (C.35) to bound the error in $\{\widehat{\mathbf{b}}^{1v}\}_{v=2}^{m}$.

Combining the three error bounds above gives us the desired result for trade-off parameter in Theorem 5.1.

QED.

APPENDIX D: QUADRATIC PERFORMANCE METRIC ELICITATION

D.1 GEOMETRY OF THE FEASIBLE SPACE (PROOFS OF SECTION 6.1.2, 6.3.2)

Proof of Proposition 6.1 and Proposition 6.2. The proof of Proposition 6.2 is same as Proposition 5.1. The proof of Proposition 6.1 is analogous where the probability measures (corresponding to classifiers and their rates) are not conditioned on any group. QED.

D.1.1 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 6.1(b)). The following discussion is borrowed from Appendix C and provided here for completeness.

The following optimization problem is a special case of OP2 in [15]. 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 $\epsilon > 0$.

$$\min_{\mathbf{r}\in\mathcal{R}} 0 \qquad s.t. \;\; \|\mathbf{r} - \mathbf{r}_0\|_2 \leq \epsilon. \tag{D.1}$$

The above optimization problem checks the feasibility, and if there exists a solution to the above problem, then Algorithm 1 of [15] returns it. Furthermore, Algorithm D.1 computes a value of $\rho \geq \widetilde{p}/k$, where $\widetilde{p}$ is the radius of the largest ball contained in the set $\mathcal{R}$. Also, the approach in [15] 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 D.1. Let $\widetilde{p}$ denote the radius of the largest ball in $\mathcal{R}$ centered at $\mathbf{o}$. Then Algorithm D.1 returns a sphere with radius $\rho \geq \widetilde{p}/k$, where k is the number of classes.

The idea in Algorithm D.1 can be trivially extended to finding a sphere $\overline{S} \subset \mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$ corresponding to Remark 6.3.

D.2 QUADRATIC PERFORMANCE METRIC ELICITATION PROCEDURE

In this section, we describe how the subroutine calls to LPME in Algorithm 6.1 elicit a quadratic metric in Definition 6.3. We start with the shifted metric of Equation (6.14).

Algorithm D.1 Obtaining the sphere $\mathcal{S} \subset \mathcal{R}$ (Figure 6.1(b)) of radius ρ centered at $\mathbf{o}$

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

As explained in Chapter 6, we may assume $d_1 \neq 0$ due to Assumption 6.2. 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 q trivial queries of the form $(\varrho\boldsymbol{\alpha}_i + \mathbf{o}, \mathbf{o}), \forall i \in [q]$.

1. From line 2 of Algorithm 6.1, we get local linear approximation at $\mathbf{o}$. Using Remark 6.2, we have (6.15) which is

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

2. Similarly, if we apply LPME on small balls around rate profiles $\mathbf{z}_j$, Remark 6.2 gives us:

$$\frac{d_i + (\rho - \varrho)B_{ij}}{d_1 + (\rho - \varrho)B_{1j}} = \frac{f_{ij}}{f_{1j}} \quad \forall\, i \in \{2, \ldots, q\},\ j \leq i. \tag{D.3}$$

$$\begin{aligned}
\implies\ & d_i + (\rho - \varrho)B_{ij} = \frac{f_{ij}}{f_{1j}}(d_1 + (\rho - \varrho)B_{1j}) \\
\implies\ & (\rho - \varrho)B_{ij} = \frac{f_{ij}}{f_{1j}}(d_1 + (\rho - \varrho)B_{j1}) - d_i \\
\implies\ & (\rho - \varrho)B_{ij} = \frac{f_{ij}}{f_{1j}}(d_1 + \frac{f_{j1}}{f_{11}}(d_1 + (\rho - \varrho)B_{11}) - d_j) - \frac{f_{i0}}{f_{10}} d_1 \\
\implies\ & (\rho - \varrho)B_{ij} = \left(\frac{f_{ij}}{f_{1j}} - \frac{f_{i0}}{f_{10}} + \frac{f_{ij}}{f_{1j}}\left(\frac{f_{j1}}{f_{11}} - \frac{f_{j0}}{f_{10}}\right)\right) d_1 + (\rho - \varrho)\frac{f_{j1}}{f_{11}} B_{11}, \qquad \text{(D.4)}
\end{aligned}$$

where we have used that the matrix $\mathbf{B}$ is symmetric in the second step, and (D.2) 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 6.2 gives us (6.17):

Algorithm D.2 Fair (Quadratic) Performance Metric Elicitation

```
**Input:** Query set $\mathcal{S}'$, search tolerance $\epsilon > 0$, oracle $\Omega'$
Let $\mathcal{L} \leftarrow \varnothing$
**for** $\sigma \in \mathcal{M}$ **do**
    $\boldsymbol{\beta}^\sigma \leftarrow \text{QPME}(\mathcal{S}', \epsilon, \Omega')$
    Let $\ell^\sigma$ be Eq. (D.11), extend $\mathcal{L} \leftarrow \mathcal{L} \cup \{\ell^\sigma\}$
**end for**
$\widehat{\mathbb{B}} \leftarrow$ normalized solution from (D.15) using $\mathcal{L}$
$\widehat{\lambda} \leftarrow$ trace back normalized solution from (D.15) for any $\sigma$
**Output:** $\widehat{\mathbf{a}}, \widehat{\mathbb{B}}, \widehat{\lambda}$
```

$$\frac{d_2 - (\rho - \varrho)B_{21}}{d_1 - (\rho - \varrho)B_{11}} = \frac{f^-_{21}}{f^-_{11}}. \tag{D.5}$$

4. Using (D.3) and (D.5), we have:

$$(\rho - \varrho)B_{11} = \frac{\frac{f^-_{21}}{f^-_{11}} + \frac{f_{21}}{f_{11}} - 2\frac{f_{20}}{f_{10}}}{\frac{f^-_{21}}{f^-_{11}} - \frac{f_{21}}{f_{11}}} d_1. \tag{D.6}$$

   Putting (D.6) in (D.4), we get:

$$\begin{aligned} B_{ij} &= \left[\frac{f_{ij}}{f_{1j}} \left(1 + \frac{f_{j1}}{f_{11}}\right) - \frac{f_{ij}}{f_{1j}}\frac{f_{j0}}{f_{10}} - \frac{f_{i0}}{f_{10}} + \frac{f_{ij}}{f_{1j}}\frac{f_{j1}}{f_{11}} \frac{\frac{f^-_{21}}{f^-_{11}} + \frac{f_{21}}{f_{11}} - 2\frac{f_{20}}{f_{10}}}{\frac{f^-_{21}}{f^-_{11}} - \frac{f_{21}}{f_{11}}} \right] d_1 \\ &= \left(F_{i,1,j}(1 + F_{j,1,1}) - F_{i,1,j}F_{j,1,0} - F_{i,1,0} + F_{i,1,j} \frac{F^-_{2,1,1} + F_{2,1,1} - 2F_{2,1,0}}{F^-_{2,1,1} - F_{2,1,1}} \right) d_1, \end{aligned} \tag{D.7}$$

   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 (D.2) and (D.7) 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 6.2).

D.3 FAIR (QUADRATIC) PERFORMANCE METRIC ELICITATION PROCEDURE

We first discuss eliciting the fair (quadratic) metric in Definition 6.4, 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 [22]. However, unlike Zhang et al. [22], 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 Chapter 6 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 6.2.

Suppose that the oracle's fair performance metric is ϕ^{fair} parametrized by $(\mathbf{a}, \mathbb{B}, \lambda)$ as in Definition 6.4. The overall fair metric elicitation procedure framework is summarized in Algorithm D.2. The framework exploits the sphere $\overline{\mathcal{S}} \subset \mathcal{R}^1 \cap \cdots \cap \mathcal{R}^m$ and uses the QPME procedure (Algorithm 6.1) 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:

$$\mathbf{r}^g = \begin{cases} \mathbf{s} & \text{if } g \in \sigma \\ \mathbf{o} & \text{o.w.} \end{cases} \tag{D.8}$$

This is an extension of the sphere $\mathcal{S}'$ defined in Chapter 6 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 6.4 reduces to:

$$\phi^{\text{fair}}(\mathbf{r}^{1:m} \in \mathcal{S}'\,; \mathbf{a}, \mathbb{B}, \lambda) = (1-\lambda)\langle \mathbf{a} \odot \boldsymbol{\tau}^\sigma, \mathbf{s} - \mathbf{o}\rangle + \lambda \frac{1}{2}(\mathbf{s} - \mathbf{o})^T \mathbf{W}^\sigma (\mathbf{s} - \mathbf{o}) + c^\sigma \tag{D.9}$$

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 (6.13) with $\mathbf{d} \coloneqq (1-\lambda)\mathbf{a} \odot \boldsymbol{\tau}^\sigma$ and $\mathbf{B} \coloneqq \lambda \mathbf{W}^\sigma$; thus, we apply QPME procedure as a subroutine in Algorithm D.2 to elicit the metric in (D.9).

The only change needed to be made to the algorithm is in line 7, 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 line 2 of Algorithm 6.1

and (6.15) an estimate

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

Using the normalization condition (i.e., $\|\mathbf{a}\|_2 = 1$), we directly get an estimate $\widehat{\mathbf{a}}$ for the linear coefficients. Similarly, steps 3-5 of Algorithm 6.1 and (6.18) gives us:$\widehat{B}_{ij} =$

$$\begin{aligned} \sum_{u \in \sigma, v \in [m] \setminus \sigma} \widetilde{B}_{ij}^{uv} &= \Big(F_{i,1,j}^\sigma (1 + F_{j,1,1}^\sigma) - F_{i,1,j}^\sigma F_{j,1,0}^\sigma d_1 - F_{i,1,0}^\sigma + F_{i,1,j}^{\sigma} {}^{\frac{F_{2,1,1}^{-,\sigma} + F_{2,1,1}^\sigma - 2F_{2,1,0}^\sigma}{F_{2,1,1}^{-,\sigma} - F_{2,1,1}^\sigma}} \Big) \tau_1^1 \widehat{a}_1 \\ &= \beta^\sigma, \end{aligned} \tag{D.11}$$

where the above solution is similar to the two group case in (6.25), but here it is corresponding to a partition of groups defined by σ, and $\widetilde{\mathbf{B}}^{uv} := \lambda \mathbf{B}^{uv}/(1 - \lambda)$ is a scaled version of the true (unknown) $\mathbf{B}^{uv}$. Let equation (D.11) be denoted by ℓ^σ. Also, let the right hand side term of (D.11) 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 (D.11). 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 5 in Algorithm D.2), which is:

$$\begin{bmatrix} \Xi & 0 & \dots & 0 \\ 0 & \Xi & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \Xi \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{b}}_{(11)} \\ \widetilde{\mathbf{b}}_{(12)} \\ \dots \\ \widetilde{\mathbf{b}}_{(qq)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\beta}_{(11)} \\ \boldsymbol{\beta}_{(12)} \\ \dots \\ \boldsymbol{\beta}_{(qq)} \end{bmatrix}, \tag{D.12}$$

where $\widetilde{\mathbf{b}}_{(ij)} = (\widetilde{b}_{ij}^1, \widetilde{b}_{ij}^2, \cdots, \widetilde{b}_{ij}^{\binom{m}{2}})$ and $\boldsymbol{\gamma}_{(ij)} = (\beta_{ij}^1, \beta_{ij}^2, \cdots, \beta_{ij}^{\binom{m}{2}})$ are vectorized versions of the ij-th entry across groups for $i, j \in [q]$, 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:

$$\Xi = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}. \tag{D.13}$$

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:

$$\begin{bmatrix} \widetilde{\mathbf{b}}_{(11)} \\ \widetilde{\mathbf{b}}_{(12)} \\ \dots \\ \widetilde{\mathbf{b}}_{(qq)} \end{bmatrix} = \begin{bmatrix} \Xi & 0 & \dots & 0 \\ 0 & \Xi & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \Xi \end{bmatrix}^{(-1)} \begin{bmatrix} \boldsymbol{\beta}_{(11)} \\ \boldsymbol{\beta}_{(12)} \\ \dots \\ \boldsymbol{\beta}_{(qq)} \end{bmatrix}. \tag{D.14}$$

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

$$\widehat{\mathbf{B}}^{uv} = \frac{\widetilde{\mathbf{B}}^{uv}}{\frac{1}{2}\sum_{u,v=1,v>u}^{m} \|\widetilde{\mathbf{B}}^{uv}\|_F} \quad \text{for} \quad u, v \in [m], v > u. \tag{D.15}$$

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

Given estimates $\widehat{B}_{ij}^{uv}$ and $\widehat{a}_1$, we can now additionally estimate the trade-off parameter $\widehat{\lambda}$ from ℓ^σ (D.11) for any $\sigma \in \mathcal{M}$. This completes the fair (quadratic) metric elicitation procedure.

D.3.1 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 Chapter 5 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. [22], 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 6.4, which then reduces the problem to a one-dimensional binary search. We have:

$$\phi^{\text{fair}}(\mathbf{r}^{1:m}\,;\,\mathbf{a}, \mathbb{B}, \lambda) \coloneqq (1-\lambda)\langle \mathbf{a}, \mathbf{r}\rangle + \lambda\frac{1}{2}\left(\sum_{u,v=1,v>u}^{m} (\mathbf{r}^u - \mathbf{r}^v)^T \mathbf{B}^{uv} (\mathbf{r}^u - \mathbf{r}^v)\right). \tag{D.16}$$

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 6.4 reduces to:

$$\phi^{\text{fair}}((\mathbf{s}, \mathbf{o}, \dots, \mathbf{o})\,;\,\mathbf{a}, \mathbb{B}, \lambda) = (1-\lambda)\langle \boldsymbol{\tau}^1 \odot \mathbf{a}, \mathbf{s} - \mathbf{o}\rangle + \lambda\frac{1}{2}(\mathbf{s}-\mathbf{o})^T \sum_{v=2}^{m} \mathbf{B}^{1v}(\mathbf{s}-\mathbf{o}) + c. \tag{D.17}$$

Algorithm D.3 Eliciting the trade-off λ when predictive performance and fairness violation are known

```
Input: Query space $\overline{\mathcal{S}}'_{\mathbf{z}_1}$, binary-search tolerance $\epsilon > 0$, oracle $\Omega$
Initialize: $\lambda^{(a)} = 0$, $\lambda^{(b)} = 1$.
while $\left|\lambda^{(b)} - \lambda^{(a)}\right| > \epsilon$ do
    Set $\lambda^{(c)} = \frac{3\lambda^{(a)}+\lambda^{(b)}}{4}$, $\lambda^{(d)} = \frac{\lambda^{(a)}+\lambda^{(b)}}{2}$, $\lambda^{(e)} = \frac{\lambda^{(a)}+3\lambda^{(b)}}{4}$
    Set $\mathbf{s}^{(a)} = \underset{\mathbf{s}\in\overline{\mathcal{S}}'_{\mathbf{z}_1}}{\operatorname{argmax}} \langle (1-\lambda^{(a)})\boldsymbol{\tau}^1 \odot \widehat{\mathbf{a}} + \lambda^{(a)} \sum_{v=2}^{m} \widehat{\mathbf{B}}^{1v}(\mathbf{z}_1 - \mathbf{o}), \mathbf{s}\rangle$ using Lemma 4.1
    Similarly, set $\mathbf{s}^{(c)}$, $\mathbf{s}^{(d)}$, $\mathbf{s}^{(e)}$, $\mathbf{s}^{(b)}$.
    Query $\Omega(\mathbf{s}^{(c)}, \mathbf{s}^{(a)})$, $\Omega(\mathbf{s}^{(d)}, \mathbf{s}^{(c)})$, $\Omega(\mathbf{s}^{(e)}, \mathbf{s}^{(d)})$, and $\Omega(\mathbf{s}^{(b)}, \mathbf{s}^{(e)})$.
    $[\lambda^{(a)}, \lambda^{(b)}] \leftarrow$ ShrinkInterval (responses) – subroutine analogous to the routine in Fig. B.1.
end while
Output: $\widehat{\lambda} = \frac{\lambda^{(a)}+\lambda^{(b)}}{2}$.
```

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 6.1(a). We may approximate the quadratic term on the right hand side above by its first order Taylor approximation as follows:

$$\begin{aligned}\phi^{\text{fair}}((\mathbf{s}, \mathbf{o}, \ldots, \mathbf{o}); \mathbf{a}, \mathbb{B}, \lambda) &\approx \phi^{\text{fair, apx}}((\mathbf{s}, \mathbf{o}, \ldots, \mathbf{o}); \mathbf{a}, \mathbb{B}, \lambda) \\ &= \langle (1-\lambda)\boldsymbol{\tau}^1 \odot \mathbf{a} + \lambda \sum_{v=2}^{m} \mathbf{B}^{1v}(\mathbf{z}_1 - \mathbf{o}), \mathbf{s}\rangle \qquad \text{(D.18)}\end{aligned}$$

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 Chapter 5 shows that the metric in (D.18) is quasiconcave in λ.

Lemma D.2. Under the regularity assumption that

$$\langle \boldsymbol{\tau}^1 \odot \mathbf{a}, \sum_{v=2}^{m} \mathbf{B}^{1v}(\mathbf{z}_1 - \mathbf{o})\rangle \neq 1, \qquad \text{(D.19)}$$

the function

$$\vartheta(\lambda) \coloneqq \max_{\mathbf{s}\in\overline{\mathcal{S}}'_{\mathbf{z}_1}} \phi^{\text{fair, apx}}((\mathbf{s}, \mathbf{o}, \ldots, \mathbf{o}); \mathbf{a}, \mathbb{B}, \lambda) \qquad \text{(D.20)}$$

is strictly quasiconcave (and therefore unimodal) in λ.

The unimodality of $\vartheta(\lambda)$ allows us to perform the one-dimensional binary search in Algorithm D.3 using the query space $\overline{\mathcal{S}}'_{\mathbf{z}_1}$, tolerance ϵ, and the oracle Ω. The binary search algorithm is same as Algorithm 5.2 and provided here for completeness.

D.4 ELICITATION GUARANTEE FOR THE QPME PROCEDURE

D.4.1 Sample complexity bounds

Recall from Definition 6.6 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. [132]) give us that with high probability over draw of D, the estimates $\widehat{\mathbf{r}}$ 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 $\Omega(\widehat{\mathbf{r}}_1, \widehat{\mathbf{r}}_2)$.

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 holds 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 6.1 for a hypothesis class $\mathcal{H}$ in which each classifier is a randomized combination of a finite number of deterministic classifiers chosen from $\bar{\mathcal{H}}$, and whose capacity is measured in terms of the Natarajan dimension [133] of $\bar{\mathcal{H}}$.

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

$$\mathcal{H} = \left\{ x \mapsto \sum_{t=1}^{T} \alpha_t h_t(x) \,\middle|\, T \in \mathbb{Z}_+, \alpha \in \Delta_T, h_1, \ldots, h_T \in \bar{\mathcal{H}} \right\}, \tag{D.21}$$

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. [132], 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).

D.4.2 Proofs

Before presenting the proof of Theorem 6.1, we re-write the LPME guarantees from [17] for linear metrics in the presence of an oracle noise parameter ϵ_Ω from Definition 6.6.

Lemma D.3 (LPME guarantees with oracle noise (Chapter 4))**.** Let the oracle Ω's metric be $\phi^{\text{lin}} = \langle \mathbf{a}, \mathbf{r} \rangle$ and its feedback noise parameter from Definition 6.6 be ϵ_Ω. Then, if the LPME procedure (Algorithm 4.2) is run using a sphere $\mathcal{S} \subset \mathcal{R}$ of radius ϱ and the binary-search tolerance ϵ, then by posing $O(q \log(1/\epsilon))$ queries it recovers coefficients $\widehat{\mathbf{a}}$ with $\|\mathbf{a} - \widehat{\mathbf{a}}\|_2 \leq O\left(\sqrt{q}(\epsilon + \sqrt{\epsilon_\Omega/\varrho})\right)$.

We will use the above result while proving Theorem 6.1.

Proof of Theorem 6.1. We first find the smoothness coefficient of the metric in Definition 6.3.

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

$$\|\nabla\phi(x) - \nabla\phi(y)\|_2 \leq L\|x - y\|_2. \tag{D.22}$$

For the metric in Definition 6.3, we have:

$$\begin{aligned} \|\nabla\phi^{\text{quad}}(x) - \nabla\phi^{\text{quad}}(y)\|_2 &= \|\mathbf{a} + \mathbf{B}\mathbf{x} - (\mathbf{a} + \mathbf{B}\mathbf{y})\|_2 \\ &\leq \|\mathbf{B}\|_2\|x - y\|_2 \\ &\leq \|\mathbf{B}\|_F\|x - y\|_2 \leq 1 \cdot \|x - y\|_2, \end{aligned} \tag{D.23}$$

where in the last step, we have used the scale invariance condition from Definition 6.3, i.e., $\|\mathbf{a}\|_2 + \|\mathbf{B}\|_F = 1$, which implies that $\|\mathbf{B}\|_F = 1 - \|\mathbf{a}\|_2 \leq 1$. Hence, the metrics in Definition 6.3 are 1-smooth.

Now, we look at the error in Taylor series approximation when we approximate the metric ϕ^{quad} in Definition 6.8 with a linear approximation. Our metric is

$$\phi^{\text{quad}}(\mathbf{r}) = \langle \mathbf{a}, \mathbf{r} \rangle + \frac{1}{2}\mathbf{r}^T\mathbf{B}\mathbf{r}. \tag{D.24}$$

We approximate it with the first order Taylor polynomial around a point $\mathbf{z}$:

$$T_1(\mathbf{r}) = \langle \mathbf{a}, \mathbf{z} \rangle + \frac{1}{2}\mathbf{z}^T\mathbf{B}\mathbf{z} + \langle \mathbf{a} + \mathbf{B}\mathbf{z}, \mathbf{r} \rangle \tag{D.25}$$

The bound on the error in this approximation is:

$$\begin{aligned} |E(\mathbf{r})| &= |\phi^{\text{quad}}(\mathbf{r}) - T_1(\mathbf{r})| \\ &= \frac{1}{2}|(\mathbf{r} - \mathbf{z})^T\Delta\phi^{\text{quad}}|_{\mathbf{c}}(\mathbf{r} - \mathbf{z})| \quad \text{(First-order Taylor approximation error)} \\ &= \frac{1}{2}|(\mathbf{r} - \mathbf{z})^T\mathbf{B}(\mathbf{r} - \mathbf{z})| \quad \text{(Hessian at any point } \mathbf{c} \text{ is the matrix } \mathbf{B}) \\ &\leq \frac{1}{2}\|\mathbf{B}\|_2\|\mathbf{r} - \mathbf{z}\|_2^2 \\ &\leq \frac{1}{2}\|\mathbf{B}\|_F\varrho^2 \leq \frac{1}{2}\varrho^2 \quad \text{(Due to the scale invariance condition)} \end{aligned} \tag{D.26}$$

So when the oracle is asked $\Omega(\mathbf{r}_1, \mathbf{r}_2) = \mathbf{1}[\phi^{\text{quad}}(\mathbf{r}_1) > \phi^{\text{quad}}(\mathbf{r}_2)]$, 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 D.3 later.

We first prove guarantees for the matrix $\mathbf{B}$ and then for the vector $\mathbf{a}$. We write Equation (6.18) in the following form assuming $d_1 = 1$ (since we normalize the coefficients at the end due to scale invariance):

$$
\begin{aligned}
B_{ij} = F_{ij} &= \left[\frac{f_{ij}}{f_{1j}} \left(1 + \frac{f_{j1}}{f_{11}} \right) - \frac{f_{ij}}{f_{1j}} \frac{f_{j0}}{f_{10}} - \frac{f_{i0}}{f_{10}} + \frac{f_{ij}}{f_{1j}} \frac{f_{j1}}{f_{11}} \frac{\frac{f^{-}_{21}}{f^{-}_{11}} + \frac{f_{21}}{f_{11}} - 2\frac{f_{20}}{f_{10}}}{\frac{f^{-}_{21}}{f^{-}_{11}} - \frac{f_{21}}{f_{11}}} \right]. \\
\implies \mathbf{B}[:, j] &= \mathbf{f}_j \left(\frac{1}{f_{1j}} + \frac{f_{j1}}{f_{1j} f_{11}} + \frac{f_{j0}}{f_{1j} f_{10}} + \frac{f_{j1}}{f_{1j} f_{11}} \left(\frac{\frac{f^{-}_{21}}{f^{-}_{11}} + \frac{f_{21}}{f_{11}} - 2\frac{f_{20}}{f_{10}}}{\frac{f^{-}_{21}}{f^{-}_{11}} - \frac{f_{21}}{f_{11}}} \right) \right) + \mathbf{f}_0 \frac{1}{f_{10}} \\
&= c_j \mathbf{f}_j + c_0 \mathbf{f}_0, \qquad \text{(D.27)}
\end{aligned}
$$

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 6.4. Notice that,

$$
\frac{\partial \mathbf{B}[:, j]}{\partial \mathbf{f}_j} = diag(\mathbf{c}'_j) \odot \mathbf{I} \quad \text{, and} \quad \frac{\partial \mathbf{B}[:, j]}{\partial \mathbf{f}_0} = diag(\mathbf{c}'_0) \odot \mathbf{I}, \qquad \text{(D.28)}
$$

where $\mathbf{c}'_j, \mathbf{c}'_0$ are vector of Lipschitz constants (bounded due to Assumption 6.4). This implies

$$
\begin{aligned}
\|\overline{\mathbf{B}}[:, j] - \widehat{\mathbf{B}}[:, j]\|_2 &\leq c'_j \|\bar{\mathbf{f}}_j - \widehat{\mathbf{f}}_j\|_2 + c'_0 \|\bar{\mathbf{f}}_0 - \widehat{\mathbf{f}}_0\|_2 \\
&\leq c'_j \sqrt{q} \left(\epsilon + \sqrt{\varrho + \epsilon_\Omega / \varrho} \right) + c'_0 \sqrt{q} \left(\epsilon + \sqrt{\varrho + \epsilon_\Omega / \varrho} \right) \\
&= O\left(\sqrt{q} \left(\epsilon + \sqrt{\varrho + \epsilon_\Omega / \varrho} \right) \right), \qquad \text{(D.29)}
\end{aligned}
$$

where we have used LPME guarantees from Lemma D.3 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 $\max_{ij} |B_{ij} - \widehat{B}_{ij}| \leq O\left(\sqrt{q} \left(\epsilon + \sqrt{\varrho + \epsilon_\Omega / \varrho} \right) \right)$, and consequentially, $\|\mathbf{B} - \widehat{\mathbf{B}}\|_F \leq O\left(q\sqrt{q} \left(\epsilon + \sqrt{\varrho + \epsilon_\Omega / \varrho} \right) \right)$.

Now let us look at guarantees for $\mathbf{a}$. Since $\mathbf{a} = \mathbf{d} - \mathbf{B}\mathbf{o}$ from (6.13), we can write

$$
\mathbf{a} = c_0 \mathbf{f}_0 - \sum_{j=1}^{q} o_j \mathbf{B}[:, j], \qquad \text{(D.30)}
$$

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

$$\frac{\partial \mathbf{a}}{\partial \mathbf{f}_0} = c_0 \mathbf{I} \quad \text{and} \quad \frac{\partial \mathbf{a}}{\partial \mathbf{B}[:, j]} = \frac{1}{k}\mathbf{I}. \tag{D.31}$$

Thus,

$$\begin{aligned} \|\mathbf{a} - \widehat{\mathbf{a}}\|_2 &\leq c_0' \sqrt{q} \left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right) + \frac{1}{k}\sum_{j=1}^{q} \sqrt{q} \left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right) \\ &= c_0' \sqrt{q} \left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right) + \frac{1}{\sqrt{q}}\sum_{j=1}^{q} c_j' \sqrt{q} \left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right) \\ &= O\left(q\left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right)\right), \end{aligned} \tag{D.32}$$

where c_0', c_j''s are some Lipschitz constants (bounded due to Assumption 6.4), and we have used the fact that $q = k^2 - k$ in the second step. QED.

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 D.2. Let ϕ^{quad} be the oracle's quadratic metric and $\widehat{\phi}^{\text{quad}}$ be its estimate obtained by the QPME procedure (Algorithm 6.1). Moreover, let $\mathbf{r}^*$ and $\widehat{\mathbf{r}}^*$ be the minimizers of ϕ^{quad} and $\widehat{\phi}^{\text{quad}}$, respectively. Then, $\phi^{\text{quad}}(\widehat{\mathbf{r}}^*) \leq \phi^{\text{quad}}(\mathbf{r}^*) + O\left(q^2\sqrt{q}\left(\epsilon + \sqrt{\varrho + \epsilon_\Omega/\varrho}\right)\right)$.

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

$$\begin{aligned} \phi^{\text{quad}}(\widehat{\mathbf{r}}^*) &\leq \widehat{\phi}^{\text{quad}}(\widehat{\mathbf{r}}^*) + \epsilon && \left(\text{as } \widehat{\phi}^{\text{quad}} \text{ approximates } \phi^{\text{quad}}\right) \\ &\leq \widehat{\phi}^{\text{quad}}(\mathbf{r}^*) + \epsilon && \left(\text{as } \widehat{\mathbf{r}}^* \text{ minimizes } \widehat{\phi}^{\text{quad}}\right) \\ &\leq \phi^{\text{quad}}(\mathbf{r}^*) + 2\epsilon && \left(\text{as } \widehat{\phi}^{\text{quad}} \text{ approximates } \phi^{\text{quad}}\right) \end{aligned} \tag{D.33}$$

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

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

where in the fourth step, we have used the fact that the rates are bounded in $[0, 1]$; hence $\|\mathbf{r}\|_2 \leq \sqrt{q}$, and in the fifth step, we have used the guarantees from Theorem 6.1. Combining(D.33) and (D.34) gives us the desired result. QED.

Proof of Theorem 6.2. 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 $O(q^2 \log(1/\epsilon)$ queries, we have $\|a - \widehat{a}\|_2 \leq O(q\epsilon)$ and $\left\|B - \widehat{B}\right\|_F \leq O(q\sqrt{q}\epsilon)$.

If we vectorize the tuple $(\mathbf{a}, \mathbf{B})$ and denote it by w, we have $\|w - \widehat{w}\|_2 \leq O(q\sqrt{q}\epsilon)$, where both $\|w\|_2, \|\widehat{w}\|_2 = 1$, due to the scale invariance condition from Definition 6.3. Note that w is $\frac{q^2+3q}{2}$-dimensional vector and defines the scale-invariant quadratic metric elicitation problem. Now, we have to count the minimum number of $\widehat{w}$ that are possible such that $\|w - \widehat{w}\|_2 \leq O(q\sqrt{q}\epsilon)$.

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

$$
\begin{aligned}
Vol(\mathbb{B}) &=\leq \sum_{i=1}^{N} Vol(q\sqrt{q}\epsilon\mathbb{B} + u_i) \\
&= NVol(q\sqrt{q}\epsilon\mathbb{B}) \\
&= (q\sqrt{q}\epsilon)^{\frac{q^2+3q}{2}-1}. \qquad \text{(D.35)}
\end{aligned}
$$

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

$$
c\left(\frac{1}{q\sqrt{q}\epsilon}\right)^{\frac{q^2+3q}{2}-1} \leq N, \qquad \text{(D.36)}
$$

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

APPENDIX E: OPTIMIZING BLACK-BOX METRICS THROUGH METRIC ELICITATION

Notation: For an index $j \in [k]$, $\text{onehot}(j) \in \{0,1\}^k$ denotes a one-hot encoding of j, and for a classifier $h : \mathcal{X} \to [k]$, $\widetilde{h} = \text{onehot}(h)$ denotes the same classifier with one-hot outputs, i.e. $\widetilde{h}(x) = \text{onehot}(h(x))$.

E.1 EXTENSION TO GENERAL LINEAR METRICS

We describe how our proposal extends to black-box metrics $\mathcal{E}^D[h] = \psi(\mathbf{C}[h])$ defined by a function $\psi : [0,1]^{k\times k} \to \mathbb{R}_+$ of *all* confusion matrix entries. This handles, for example, the label noise models in Table 7.1 with a general (non-diagonal) noise transition matrix $\mathbf{T}$. We begin with metrics that are linear functions of the diagonal and off-diagonal confusion matrix entries $\mathcal{E}^D[h] = \sum_{ij} \beta_{ij} C_{ij}[h]$ for some $\boldsymbol{\beta} \in \mathbb{R}^{k\times k}$. In this case, we will use an example weighting function $\mathbf{W} : \mathcal{X} \to \mathbb{R}_+^{k\times k}$ that maps an instance x to an $k \times k$ weight matrix $\mathbf{W}(x)$, where $W_{ij}(x) \in \mathbb{R}_+^{k\times k}$ is the weight associated with the (i,j)-th confusion matrix entry.

Note that in practice, the metric $\mathcal{E}^D$ may depend on only a subset of d entries of the confusion matrix, in which case, the weighting function only needs to weight those entries. Consequently, the weighting function can be parameterized with Ld parameters, which can then be estimated by solving a system of Ld linear equations. For the sake of completeness, here we describe our approach for metrics that depend on all k^2 confusion entries.

Modeling weighting function: Like in (7.7), we propose modeling this function as a weighted sum of L basis functions:

$$W_{ij}(x) \,=\, \sum_{\ell=1}^{L} \alpha_{ij}^{\ell} \phi^{\ell}(x), \tag{E.1}$$

where each $\phi^\ell : \mathcal{X} \to [0,1]$ and $\alpha_{ij}^\ell \in \mathbb{R}$. Similar to (7.6), our goal is to then estimate coefficients $\boldsymbol{\alpha}$ so that:

$$\mathbf{E}_{(x,y)\sim\mu}\Big[\sum_{ij} W_{ij}(x)\,\mathbf{1}(y=i)h_j(x)\Big] \,\approx\, \mathcal{E}^D[h], \forall h. \tag{E.2}$$

Expanding the weighting function in (E.2), we get:

$$\sum_{\ell=1}^{L}\sum_{i,j} \alpha_{ij}^{\ell}\, \underbrace{\mathbf{E}_{(x,y)\sim\mu}\big[\phi^\ell(x)\,\mathbf{1}(y=i)h_j(x)\big]}_{\Phi^{\mu,\ell}_{i,j}[h]} \,\approx\, \mathcal{E}^D[h], \forall h, \tag{E.3}$$

which can be re-written as:

$$\sum_{\ell=1}^{L}\sum_{i,j}\alpha^{\ell}_{ij}\Phi^{\mu,\ell}_{ij}[h] \;\approx\; \mathcal{E}^{D}[h], \forall h. \tag{E.4}$$

Estimating coefficients $\boldsymbol{\alpha}$: To estimate $\boldsymbol{\alpha} \in \mathbb{R}^{Lk^2}$, our proposal is to probe the metric $\mathcal{E}^D$ at Lk^2 different classifiers $h^{\ell,1,1}, \ldots, h^{\ell,k,k}$, with one classifier for each combination (ℓ, i, j) of basis functions and confusion matrix entries, and to solve the following system of Lk^2 linear equations:

$$\begin{aligned}\sum_{\ell,i,j}\alpha^{\ell}_{ij}\,\widehat{\Phi}^{\mathrm{tr},\ell}_{ij}[h^{1,1,1}] &= \widehat{\mathcal{E}}^{\mathrm{val}}[h^{1,1,1}]\\ &\vdots\\ \sum_{\ell,i,j}\alpha^{\ell}_{ij}\,\widehat{\Phi}^{\mathrm{tr},\ell}_{ij}[h^{L,m,m}] &= \widehat{\mathcal{E}}^{\mathrm{val}}[h^{L,k,k}]\end{aligned} \tag{E.5}$$

Here $\widehat{\Phi}^{\mathrm{tr},\ell}_{ij}[h]$ is an estimate of $\Phi^{\mu,\ell}_{ij}[h]$ using training sample S^{tr} and $\widehat{\mathcal{E}}^{\mathrm{val}}[h]$ is an estimate of $\mathcal{E}^{D}[h]$ using the validation sample S^{val}. Equivalently, defining $\widehat{\boldsymbol{\Sigma}} \in \mathbb{R}^{Lk^2 \times Lk^2}$ and $\widehat{\boldsymbol{\mathcal{E}}} \in \mathbb{R}^{Lk^2}$ with each:

$$\widehat{\Sigma}_{(\ell,i,j),(\ell',i',j')} = \widehat{\Phi}^{\mathrm{tr},\ell'}_{i'j'}[h^{\ell,i,j}]; \quad \widehat{\mathcal{E}}_{(\ell,i,j)} = \widehat{\mathcal{E}}^{\mathrm{val}}[h^{\ell,i,j}], \tag{E.6}$$

we compute $\widehat{\boldsymbol{\alpha}} = \widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\mathcal{E}}}$.

Choosing probing classifiers: As described in Section 7.3.4, we propose picking each probing classifier $h^{\ell,i,j}$ so that the (ℓ, i, j)-th diagonal entry of $\widehat{\boldsymbol{\Sigma}}$ is large and the off-diagonal entries are all small. This can be framed as the following constrained satisfaction problem:

$$\text{For } h^{\ell,i,j} \text{ pick } h \in \mathcal{H} \text{ such that:}$$

$$\widehat{\Phi}^{\mathrm{tr},\ell}_{i,j}[h] \geq \gamma, \text{ and } \widehat{\Phi}^{\mathrm{tr},\ell'}_{i',j'}[h] \leq \omega, \forall(\ell', i', j') \neq (\ell, i, j), \tag{E.7}$$

for some $0 < \omega < \gamma < 1$. While the more practical approach prescribed in Section 7.3.4 of constructing the probing classifiers from trivial classifiers that predict the same class on all or a subset of examples does not apply here (because here we need to take into account both the diagonal and off-diagonal confusion entries), the above problem can be solved using off-the-shelf tools available for rate-constrained optimization problems [105].

Plug-in classifier: Having estimated an example weighting function $\widehat{\mathbf{W}} : \mathcal{X} \to \mathbb{R}^{k \times k}$, we seek to maximize a weighted objective on the training distribution:

$$\max_{h} \mathbf{E}_{(x,y)\sim\mu}\left[\sum_{ij} \widehat{W}_{ij}(x)\,\mathbf{1}(y=i)h_j(x)\right], \tag{E.8}$$

for which we can construct a plug-in classifier that post-shifts a pre-trained class probability model $\widehat{\eta}^{\text{tr}} : \mathcal{X}{\to}\Delta_k$:

$$\widehat{h}(x) \in \operatorname*{argmax}_{j\in[k]} \sum_{i=1}^{k} \widehat{W}_{ij}(x)\,\widehat{\eta}_i^{\text{tr}}(x). \tag{E.9}$$

For handling general non-linear metrics $\mathcal{E}^D[h] = \psi(\mathbf{C}[h])$ with a smooth $\psi : [0,1]^{k\times k}{\to}\mathbb{R}_+$, we can directly adapt the iterative plug-in procedure in Algorithm 7.3, which would in turn construct a plug-in classifier of the above form in each iteration (line 9). See [18] for more details of the iterative Frank-Wolfe based procedure for optimizing general metrics, where the authors consider non-black-box metrics in the absence of distribution shift.

E.2 PROOFS

E.2.1 Proof of Theorem 7.1

Theorem E.1 ((Restated) **Error bound on elicited weights).** Let the input metric be of the form $\widehat{\mathcal{E}}^{\text{lin}}[h] = \sum_i \beta_i \widehat{C}_{ii}^{\text{val}}[h]$ for some (unknown) coefficients $\boldsymbol{\beta} \in \mathbb{R}_+^k$, $\|\boldsymbol{\beta}\| \leq 1$. Let $\mathcal{E}^D[h] = \sum_i \beta_i C_{ii}^D[h]$. Let $\gamma, \omega > 0$ be such that the constraints in (7.14) are feasible for hypothesis class $\bar{\mathcal{H}}$, for all ℓ, i. Suppose Algorithm 7.1 chooses each classifier $h^{\ell,i}$ to satisfy (7.14), with $\mathcal{E}^D[h^{\ell,i}] \in [c,1], \forall \ell, i,$ for some $c > 0$. Let $\bar{\alpha}$ be the associated coefficient in Assumption 7.1 for metric $\mathcal{E}^D$. Suppose $\gamma > 2\sqrt{2}Lk\omega$ and $n^{\text{tr}} \geq \frac{L^2 k \log(Lk|\mathcal{H}|/\delta)}{(\frac{\gamma}{2}-\sqrt{2}Lk\omega)^2}$. Fix $\delta \in (0,1)$. Then w.p. $\geq 1-\delta$ over draws of S^{tr} and S^{val} from μ and D resp., the coefficients $\widehat{\boldsymbol{\alpha}}$ output by Algorithm 7.1 satisfies:

$$\|\widehat{\boldsymbol{\alpha}} - \bar{\boldsymbol{\alpha}}\| \leq \mathcal{O}\Big(\frac{Lk}{\gamma^2}\sqrt{\frac{L\log(Lk|\mathcal{H}|/\delta)}{n^{\text{tr}}}} + \frac{\sqrt{Lk}}{\gamma}\Big(\sqrt{\frac{L^2 k\log(Lk/\delta)}{c^2\gamma^2 n^{\text{val}}}} + \nu\Big)\Big), \tag{E.10}$$

where the term $|\mathcal{H}|$ can be replaced by a measure of capacity of the hypothesis class $\mathcal{H}$.

The solution from Algorithm 7.1 is given by $\widehat{\boldsymbol{\alpha}} = \widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\mathcal{E}}}$. Let $\bar{\boldsymbol{\alpha}}$ be the "true" coefficients given in Assumption 7.1. Let $\boldsymbol{\Sigma} \in \mathbb{R}^{Lk\times Lk}$ denote the population version of $\widehat{\boldsymbol{\Sigma}}$, with $\Sigma_{(\ell,i),(\ell',i')} = \mathbf{E}_{(x,y)\sim\mu}\big[\phi^{\ell'}(x)\mathbf{1}(y=i')h_{i'}^{\ell,i}(x)\big]$. Similarly, denote the population version of $\widehat{\boldsymbol{\mathcal{E}}}$ by: $\mathcal{E}_{(\ell,i)} = \mathcal{E}^D[h^{\ell,i}]$. Let $\boldsymbol{\alpha} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mathcal{E}}$ be the solution we obtain had we used the population versions of these quantities. Further, define the vector $\bar{\boldsymbol{\mathcal{E}}} \in \mathbb{R}^{Lk}$:

$$\bar{\mathcal{E}}_{(\ell',i')} = \sum_{\ell,i} \bar{\alpha}_i^{\ell} \Phi_i^{\mu,\ell}[h^{\ell',i'}]. \tag{E.11}$$

It trivially follows that the coefficient $\bar{\boldsymbol{\alpha}}$ given by Assumption 7.1 can be written as $\bar{\boldsymbol{\alpha}} = \boldsymbol{\Sigma}^{-1}\bar{\boldsymbol{\mathcal{E}}}$.

We will find the following lemmas useful. Our first two lemmas bound the gap between the empirical and population versions of $\boldsymbol{\Sigma}$ (the left-hand side of the linear system) and $\boldsymbol{\mathcal{E}}$ (the right-hand side of the linear system).

Lemma E.1 (Confidence bound for $\boldsymbol{\Sigma}$)**.** Fix $\delta \in (0,1)$. With probability at least $1-\delta$ over draw of S^{tr} from μ,

$$|\Sigma_{(\ell,i),(\ell',i')} - \widehat{\Sigma}_{(\ell,i),(\ell',i')}| \le \mathcal{O}\left(\sqrt{\frac{p_{\ell,i}\log(Lk|\mathcal{H}|/\delta)}{n^{\text{tr}}}}\right), \tag{E.12}$$

where $p_{\ell,i} = \mathbf{E}_{(x,y)\sim\mu}[\phi^{\ell}(x)\mathbf{1}(y=i)]$, and consequently,

$$\|\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}\| \le \mathcal{O}\left(\sqrt{\frac{L^2 k\log(Lk|\mathcal{H}|/\delta)}{n^{\text{tr}}}}\right). \tag{E.13}$$

Proof. Each row of $\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}$ contains the difference between the elements $\Phi_i^{\mu,\ell}[h]$ and $\widehat{\Phi}_i^{\text{tr},\ell}[h]$ for a classifier h chosen from $\mathcal{H}$. Using multiplicative Chernoff bounds, we have for a fixed h, with probability at least $1-\delta$ over draw of S^{tr} from μ

$$|\Phi_i^{\mu,\ell}[h] - \widehat{\Phi}_i^{\text{tr},\ell}[h]| \le \mathcal{O}\left(\sqrt{\frac{p_{\ell,i}\log(1/\delta)}{n^{\text{tr}}}}\right), \tag{E.14}$$

where $p_{\ell,i} = \mathbf{E}_{(x,y)\sim\mu}[\phi^{\ell}(x)\mathbf{1}(y=i)]$. Taking a union bound over all $h \in \mathcal{H}$, we have with probability at least $1-\delta$ over draw of S^{tr} from μ, for any $h \in \mathcal{H}$:

$$|\Phi_i^{\mu,\ell}[h] - \widehat{\Phi}_i^{\text{tr},\ell}[h]| \le \mathcal{O}\left(\sqrt{\frac{p_{\ell,i}\log(|\mathcal{H}|/\delta)}{n^{\text{tr}}}}\right). \tag{E.15}$$

Taking a union bound over all $Lk \times Lk$ entries, we have with probability at least $1-\delta$, for all $(\ell,i),(\ell',i')$:

$$|\Sigma_{(\ell,i),(\ell',i')} - \widehat{\Sigma}_{(\ell,i),(\ell',i')}| \le \mathcal{O}\left(\sqrt{\frac{p_{\ell,i}\log(Lk|\mathcal{H}|/\delta)}{n^{\text{tr}}}}\right). \tag{E.16}$$

Upper bounding the operator norm of $\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}$ with the Frobenius norm, we have

$$\begin{aligned}\|\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}\| &\leq \mathcal{O}\left(\sqrt{\frac{\log(Lk|\mathcal{H}|/\delta)}{n^{\text{tr}}}}\sqrt{\sum_{(\ell,i),(\ell',i')} p_{\ell',i'}}\right)\\ &\leq \mathcal{O}\left(\sqrt{\frac{\log(Lk|\mathcal{H}|/\delta)}{n^{\text{tr}}}}\sqrt{\sum_{\ell,i,\ell'}(1)}\right) \leq \mathcal{O}\left(\sqrt{\frac{L^2k\log(Lk|\mathcal{H}|/\delta)}{n^{\text{tr}}}}\right), \qquad \text{(E.17)}\end{aligned}$$

where the second inequality uses the fact that $\sum_{i'} p_{\ell',i'} = \mathbf{E}_{x\sim\mathbf{P}^{\mu}}\left[\phi^{\ell'}(x)\right] \leq 1$. QED.

Lemma E.2 (Confidence bound for $\boldsymbol{\mathcal{E}}$)**.** Fix $\delta \in (0,1)$. With probability at least $1-\delta$ over draw of S^{val} from D,

$$\|\boldsymbol{\mathcal{E}} - \widehat{\boldsymbol{\mathcal{E}}}\| \leq \mathcal{O}\left(\sqrt{\frac{Lk\log(Lk/\delta)}{n^{\text{val}}}}\right). \qquad \text{(E.18)}$$

Proof. From an application of Hoeffding's inequality, we have for any fixed $h^{\ell,i}$:

$$\begin{aligned}|\mathcal{E}_{(\ell,i)} - \widehat{\mathcal{E}}_{(\ell,i)}| = |\mathcal{E}^D[h^{\ell,i}] - \widehat{\mathcal{E}}^{\text{val}}[h^{\ell,i}]| &= \left|\sum_i \beta_i C^D_{ii}[h^{\ell,i}] - \sum_i \beta_i \widehat{C}^{\text{val}}_{ii}[h^{\ell,i}]\right|\\ &\leq \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{n^{\text{val}}}}\right), \qquad \text{(E.19)}\end{aligned}$$

which holds with probability at least $1-\delta$ over draw of S^{val} and uses the fact that each β_i and $C^D_{ii}[h]$ is bounded. Taking a union bound over all Lk probing classifiers, we have:

$$\|\boldsymbol{\mathcal{E}} - \widehat{\boldsymbol{\mathcal{E}}}\| \leq \mathcal{O}\left(\sqrt{Lk}\sqrt{\frac{\log(Lk/\delta)}{n^{\text{val}}}}\right). \qquad \text{(E.20)}$$

Note that we do not need a uniform convergence argument like in Lemma E.1 as the probing classifiers are chosen independent of the validation sample. QED.

Our last two lemmas show that $\boldsymbol{\Sigma}$ is well-conditioned. We first show that because the probing classifiers $h^{\ell,i}$'s are chosen to satisfy (7.14), the diagonal and off-diagonal entries of $\boldsymbol{\Sigma}$ can be lower and upper bounded respectively as follows.

Lemma E.3 (Bounds on diagonal and off-diagonal entries of $\boldsymbol{\Sigma}$)**.** Fix $\delta \in (0,1)$. With probability at least $1-\delta$ over draw of S^{tr} from μ,

$$\Sigma_{(\ell,i),(\ell,i)} \geq \gamma - \mathcal{O}\left(\sqrt{\frac{p_{\ell,i}\log(Lk|\mathcal{H}|/\delta)}{n^{\mathrm{tr}}}}\right), \forall(\ell,i) \tag{E.21}$$

and

$$\Sigma_{(\ell,i),(\ell',i')} \leq \omega + \mathcal{O}\left(\sqrt{\frac{p_{\ell,i}\log(Lk|\mathcal{H}|/\delta)}{n^{\mathrm{tr}}}}\right), \forall(\ell,i) \neq (\ell',i'), \tag{E.22}$$

where $p_{\ell,i} = \mathbf{E}_{(x,y)\sim\mu}[\phi^{\ell}(x)\mathbf{1}(y=i)]$.

Proof. Because the probing classifiers $h^{\ell,i}$'s are chosen from $\mathcal{H}$ to satisfy (7.14), we have $\widehat{\Sigma}_{(\ell,i),(\ell,i)} \geq \gamma, \forall(\ell,i)$ and $\widehat{\Sigma}_{(\ell,i),(\ell',i')} \leq \omega, \forall(\ell,i) \neq (\ell',i')$. The proof follows from generalization bounds similar to Lemma E.1. QED.

The bounds on the diagonal and off-diagonal entries of $\mathbf{\Sigma}$ then allow us to bound its smallest and largest singular values.

Lemma E.4 (Bounds on singular values of $\mathbf{\Sigma}$)**.** We have $\|\mathbf{\Sigma}\| \leq L\sqrt{k}$. Fix $\delta \in (0,1)$. Suppose $\gamma > 2\sqrt{2}Lk\omega$ and $n^{\mathrm{tr}} \geq \frac{L^2 k\log(Lk|\mathcal{H}|/\delta)}{(\frac{\gamma}{2}-\sqrt{2}Lk\omega)^2}$. With probability at least $1-\delta$ over draw of S^{tr} from μ, $\|\mathbf{\Sigma}^{-1}\| \leq \mathcal{O}\left(\frac{1}{\gamma}\right)$.

Proof. We first derive a straight-forward upper bound on the the operator norm of $\mathbf{\Sigma}$ in terms of its Frobenius norm: $\|\mathbf{\Sigma}\| \leq$

$$\sqrt{\sum_{(\ell,i),(\ell',i')} \Sigma^2_{(\ell,i),(\ell',i')}} \leq \sqrt{\sum_{(\ell,i),(\ell',i')} p^2_{\ell',i'}} \leq \sqrt{\sum_{(\ell,i),(\ell',i')} p_{\ell',i'}} \leq \sqrt{\sum_{\ell,i,\ell'} 1} = L\sqrt{k}, \tag{E.23}$$

where $p_{\ell,i} = \mathbf{E}_{(x,y)\sim\mu}[\phi^{\ell}(x)\mathbf{1}(y=i)]$ and the last inequality uses the fact that $\sum_{i'} p_{\ell',i'} = \mathbf{E}_{x\sim\mathbf{P}^{\mu}}\left[\phi^{\ell'}(x)\right] \leq 1$.

To bound the operator norm of $\|\mathbf{\Sigma}^{-1}\|$, denote $\upsilon_{\ell,i} = \mathcal{O}\left(\sqrt{\frac{p_{\ell,i}\log(Lk|\mathcal{H}|/\delta)}{n^{\mathrm{tr}}}}\right)$. From Lemma E.3, we can express $\mathbf{\Sigma}$ as a sum of a matrix $\mathbf{A}$ and a diagonal matrix $\mathbf{D}$, i.e. $\mathbf{\Sigma} = \mathbf{A} + \mathbf{D}$, where each $A_{(\ell,i),(\ell,i)} = 0$, $A_{(\ell,i),(\ell',i')} \leq \omega + \upsilon_{\ell,i}, \forall(\ell,i) \neq (\ell',i')$ and $D_{(\ell,i),(\ell,i)} \geq \gamma - \upsilon_{\ell,i}$. Let $\sigma_{\ell,i}(\mathbf{\Sigma})$ denote the (ℓ,i)-th largest singular value of $\mathbf{\Sigma}$. By Weyl's inequality, we have that the singular values of $\mathbf{\Sigma}$ can be bounded in terms of the singular values $\mathbf{D}$ (see e.g., [134]):

$$|\sigma_{\ell,i}(\mathbf{\Sigma}) - \sigma_{\ell,i}(\mathbf{D})| \leq \|\mathbf{A}\|, \quad \text{or} \quad \sigma_{\ell,i}(\mathbf{D}) - \sigma_{\ell,i}(\mathbf{\Sigma}) \leq \|\mathbf{A}\|. \tag{E.24}$$

We further have:

$$\begin{aligned}
\sigma_{\ell,i}(\mathbf{D}) - \sigma_{\ell,i}(\mathbf{\Sigma}) &\leq \|\mathbf{A}\| \leq \sqrt{\sum_{(\ell,i)\neq(\ell',i')} (\omega + \upsilon_{\ell,i})^2} + \upsilon_{\ell,i} \\
&\leq \sqrt{2}\sqrt{\sum_{(\ell,i)\neq(\ell',i')} \omega^2 + \sum_{(\ell,i)\neq(\ell',i')} \upsilon_{\ell,i}^2} + \upsilon_{\ell,i} \\
&\leq \sqrt{2}\sqrt{\sum_{(\ell,i)\neq(\ell',i')} \omega^2} + \sqrt{2}\sqrt{\sum_{(\ell,i)\neq(\ell',i')} \upsilon_{\ell,i}^2} \\
&\leq \sqrt{2}Lk\omega \,+\, \mathcal{O}\left(\sqrt{\frac{\log(Lk|\mathcal{H}|/\delta)}{n^{\mathrm{tr}}}}\right)\sqrt{\sum_{(\ell,i)\neq(\ell',i')} p_{\ell,i}} \\
&\leq \sqrt{2}Lk\omega \,+\, \mathcal{O}\left(\sqrt{\frac{L^2 k\log(Lk|\mathcal{H}|/\delta)}{n^{\mathrm{tr}}}}\right). \qquad \text{(E.25)}
\end{aligned}$$

Since $\sigma_{\ell,i}(\mathbf{D}) \geq \gamma - \max_{\ell,i}\upsilon_{\ell,i}$, and

$$\sigma_{\ell,i}(\mathbf{\Sigma}) \,\geq\, \gamma \,-\, \sqrt{2}Lk\omega \,-\, \mathcal{O}\left(\sqrt{\frac{L^2 k\log(Lk|\mathcal{H}|/\delta)}{n^{\mathrm{tr}}}}\right) - \max_{\ell,i}\upsilon_{\ell,i}. \qquad \text{(E.26)}$$

Substituting for $\max_{\ell,i}\upsilon_{\ell,i} \leq \mathcal{O}\left(\sqrt{\frac{\log(Lk|\mathcal{H}|/\delta)}{n^{\mathrm{tr}}}}\right)$, and denoting $\sqrt{2}Lk\omega + \mathcal{O}\left(\sqrt{\frac{L^2k\log(Lk|\mathcal{H}|/\delta)}{n^{\mathrm{tr}}}}\right)$ by ξ, we have $\sigma_{\ell,i}(\mathbf{\Sigma}) \geq \xi$. With this, we can bound operator norm of $\|\mathbf{\Sigma}^{-1}\|$ as:

$$\|\mathbf{\Sigma}^{-1}\| = \frac{1}{\min_{\ell,i}\sigma_{\ell,i}(\mathbf{\Sigma})} \,\leq\, \frac{1}{\gamma-\xi} \,\leq\, \mathcal{O}\left(\frac{1}{\gamma}\right), \qquad \text{(E.27)}$$

where the last inequality follows from the assumption that $n^{\mathrm{tr}} \geq \frac{L^2k\log(Lk|\mathcal{H}|/\delta)}{(\frac{\gamma}{2}-\sqrt{2}Lk\omega)^2}$ and hence $\xi \leq \mathcal{O}\left(\gamma/2\right)$. QED.

We are now ready to prove Theorem 7.1.

Proof of Theorem 7.1. The solution from Algorithm 7.1 is given by $\widehat{\boldsymbol{\alpha}} = \widehat{\mathbf{\Sigma}}^{-1}\widehat{\boldsymbol{\mathcal{E}}}$. Recall we can write the "true" coefficients by $\bar{\boldsymbol{\alpha}} = \mathbf{\Sigma}^{-1}\bar{\boldsymbol{\mathcal{E}}}$, where $\bar{\boldsymbol{\mathcal{E}}}$ is defined in (E.11), and we also defined $\boldsymbol{\alpha} = \mathbf{\Sigma}^{-1}\boldsymbol{\mathcal{E}}$. The left-hand side of Theorem 7.1 can then be expanded as:

$$\begin{aligned}
\|\widehat{\boldsymbol{\alpha}} - \bar{\boldsymbol{\alpha}}\| &\leq \|\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\| + \|\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}\| && \text{(E.28)} \\
&\leq \|\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\| + \|\mathbf{\Sigma}^{-1}(\boldsymbol{\mathcal{E}} - \bar{\boldsymbol{\mathcal{E}}})\| && \text{(E.29)} \\
&\leq \|\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\| + \|\mathbf{\Sigma}^{-1}\|\|(\boldsymbol{\mathcal{E}} - \bar{\boldsymbol{\mathcal{E}}})\| && \text{(E.30)}
\end{aligned}$$

$$\le \|\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\| + \nu\sqrt{Lk}\|\boldsymbol{\Sigma}^{-1}\| \tag{E.31}$$

$$\le \|\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\| + \frac{2\nu\sqrt{Lk}}{\gamma}. \tag{E.32}$$

The second-last step follows from Assump. 7.1, particularly, from $\left|\sum_{\ell,i} \bar{\alpha}_i^\ell \Phi_i^{\mu,\ell}[h] - \mathcal{E}^D[h]\right| \le \nu, \forall h$, which gives us that $\left|\sum_{\ell,i} \bar{\alpha}_i^\ell \Phi_i^{\mu,\ell}[h^{\ell',i'}] - \mathcal{E}^D[h^{\ell',i'}]\right| \le \nu$, for all ℓ', i'. The last step follows from Lemma E.4 and holds with probability at least $1-\delta$ over draw of S^{tr}.

All that remains is to bound the term $\|\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\|$. Given that $\widehat{\boldsymbol{\alpha}} = \widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\mathcal{E}}}$. and $\boldsymbol{\alpha} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mathcal{E}}$, we can use standard error analysis for linear systems (see e.g., [135]) to bound:

$$\begin{aligned}
\|\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\| &\le \|\boldsymbol{\alpha}\|\|\boldsymbol{\Sigma}\|\|\boldsymbol{\Sigma}^{-1}\| \left(\frac{\|\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}\|}{\|\boldsymbol{\Sigma}\|} + \frac{\|\boldsymbol{\mathcal{E}} - \widehat{\boldsymbol{\mathcal{E}}}\|}{\|\boldsymbol{\mathcal{E}}\|}\right) \\
&\le \|\boldsymbol{\Sigma}^{-1}\|^2\|\boldsymbol{\mathcal{E}}\| \left(\|\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}\| + \|\boldsymbol{\Sigma}\|\frac{\|\boldsymbol{\mathcal{E}} - \widehat{\boldsymbol{\mathcal{E}}}\|}{\|\boldsymbol{\mathcal{E}}\|}\right) \quad (\text{from } \boldsymbol{\alpha} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mathcal{E}}) \\
&\le \|\boldsymbol{\Sigma}^{-1}\|^2\|\boldsymbol{\mathcal{E}}\| \left(\|\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}\| + L\sqrt{k}\frac{\|\boldsymbol{\mathcal{E}} - \widehat{\boldsymbol{\mathcal{E}}}\|}{\|\boldsymbol{\mathcal{E}}\|}\right) \quad (\text{from Lemma E.4}) \\
&\le \|\boldsymbol{\Sigma}^{-1}\|^2\sqrt{Lk} \left(\|\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}\| + \frac{L\sqrt{k}}{\sqrt{Lk}c}\|\boldsymbol{\mathcal{E}} - \widehat{\boldsymbol{\mathcal{E}}}\|\right) \quad (\text{using } \mathcal{E}_{(\ell,i)} \in (c, 1]) \\
&\le \|\boldsymbol{\Sigma}^{-1}\|^2\sqrt{Lk} \left(\|\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}\| + \frac{\sqrt{L}}{c}\|\boldsymbol{\mathcal{E}} - \widehat{\boldsymbol{\mathcal{E}}}\|\right) \\
&\le \mathcal{O}\left(\frac{\sqrt{Lk}}{\gamma^2}\left(\sqrt{\frac{L^2k\log(Lk|\mathcal{H}|/\delta)}{n^{\mathrm{tr}}}} + \frac{\sqrt{L}}{c}\sqrt{\frac{Lk\log(Lk/\delta)}{n^{\mathrm{val}}}}\right)\right) \\
&= \mathcal{O}\left(\frac{Lk}{\gamma^2}\left(\sqrt{\frac{L\log(Lk|\mathcal{H}|/\delta)}{n^{\mathrm{tr}}}} + \frac{1}{c}\sqrt{\frac{L\log(Lk/\delta)}{n^{\mathrm{val}}}}\right)\right), \qquad \text{(E.33)}
\end{aligned}$$

where the last two steps follow from Lemmas E.1–E.2 and Lemma E.4, and hold with probability at least $1-\delta$ over draws of S^{tr} and S^{val}. Plugging this back into (E.32) completes the proof. QED.

E.2.2 Error Bound for PI-EW

We will first provide error bound for the PI-EW algorithm, which is a special case of the FW-EG algorithm. When the metric is linear, we have the following bound on the gap between the metric value achieved by classifier $\widehat{h}$ output by Algorithm 7.2, and the optimal value. This result will then be useful in proving an error bound for the FW-EG procedure (Algorithm 7.3) in the next section, that essentially focuses on the non-linear metric optimization.

Lemma E.5 (Error Bound for PI-EW). Let the input metric be of the form $\widehat{\mathcal{E}}^{\text{lin}}[h] = \sum_i \beta_i \widehat{C}^{\text{val}}_{ii}[h]$ for some (unknown) coefficients $\boldsymbol{\beta} \in \mathbb{R}^k_+, \|\boldsymbol{\beta}\| \le 1$, and denote $\mathcal{E}^{\text{lin}}[h] = \sum_i \beta_i C^D_{ii}[h]$. Let $\bar{\boldsymbol{\alpha}}$ be the associated weighting coefficient for $\mathcal{E}^{\text{lin}}$ in Assumption 7.1, with $\|\bar{\boldsymbol{\alpha}}\|_1 \le B$ and with slack ν. Fix $\delta > 0$. Suppose w.p. $\ge 1-\delta$ over draw of S^{tr} and S^{val}, the weight elicitation routine in line 2 of Algorithm 7.2 provides coefficients $\widehat{\boldsymbol{\alpha}}$ with $\|\widehat{\boldsymbol{\alpha}} - \bar{\boldsymbol{\alpha}}\| \le \kappa(\delta, n^{\text{tr}}, n^{\text{val}})$, for some function $\kappa(\cdot) > 0$. Let $B' = B + \sqrt{Lk}\,\kappa(\delta, n^{\text{tr}}, n^{\text{val}})$. Then with the same probability, the classifier $\widehat{h}$ output by Algorithm 7.2 satisfies:

$$\max_h \mathcal{E}^{\text{lin}}[h] - \mathcal{E}^{\text{lin}}[\widehat{h}] \le B'\mathbf{E}_x\left[\|\eta^{\text{tr}}(x) - \widehat{\eta}^{\text{tr}}(x)\|_1\right] \,+\, 2\sqrt{Lk}\,\kappa(\delta, n^{\text{tr}}, n^{\text{val}}) \,+\, 2\nu, \quad \text{(E.34)}$$

where $\eta^{\text{tr}}_i(x) = \mathbf{P}^\mu(y = i|x)$. Furthermore, when the metric coefficients $\|\boldsymbol{\beta}\| \le Q$, for some $Q > 0$, then

$$\max_h \mathcal{E}^{\text{lin}}[h] - \mathcal{E}^{\text{lin}}[\widehat{h}] \le Q\left(B'\mathbf{E}_x\left[\|\eta^{\text{tr}}(x) - \widehat{\eta}^{\text{tr}}(x)\|_1\right] \,+\, 2\sqrt{Lk}\,\kappa(\delta, n^{\text{tr}}, n^{\text{val}}) \,+\, 2\nu\right) \text{(E.35)}$$

Proof. For the proof, we will treat $\widehat{h}$ as a classifier that outputs one-hot labels, i.e. as classifier $\widehat{h} : \mathcal{X} \to \{0,1\}^k$ with

$$\widehat{h}(x) \,=\, \text{onehot}\left(\underset{i\in[k]}{\text{argmax}^*}\, \widehat{W}_i(x)\widehat{\eta}^{\text{tr}}_i(x)\right), \quad \text{(E.36)}$$

where argmax* breaks ties in favor of the largest class.

Let $\bar{W}_i(x) = \sum_{\ell=1}^L \bar{\alpha}^\ell_i \phi^\ell(x)$ and $\widehat{W}_i(x) = \sum_{\ell=1}^L \widehat{\alpha}^\ell_i \phi^\ell(x)$. It is easy to see that

$$|\bar{W}_i(x) - \widehat{W}_i(x)| \le \|\bar{\boldsymbol{\alpha}} - \widehat{\boldsymbol{\alpha}}\| \sqrt{\sum_{\ell=1}^L \phi^\ell(x)^2} \le \sqrt{Lk}\|\bar{\boldsymbol{\alpha}} - \widehat{\boldsymbol{\alpha}}\| \le \sqrt{Lk}\kappa, \quad \text{(E.37)}$$

where in the second inequality we use $|\phi^\ell(x)| \le 1$, and in the last inequality, we have shortened the notation $\kappa(\delta, n^{\text{tr}}, n^{\text{val}})$ to κ and for simplicity will avoid mentioning that this holds with high probability.

Further, recall from Assumption 7.1 that

$$|\bar{W}_i(x)| \le \|\bar{\boldsymbol{\alpha}}\|_1 \max_\ell |\phi^\ell(x)| \le B(1) = B \quad \text{(E.38)}$$

and so from (E.37),

$$|\widehat{W}_i(x)| \le B + \sqrt{Lk}\kappa. \tag{E.39}$$

We also have from Assumption 7.1 that

$$\left|\mathcal{E}^{\mathrm{lin}}[h] \,-\, \mathbf{E}_{(x,y)\sim\mu}\left[\sum_{i=1}^{k}\bar{W}_i(x)\mathbf{1}(y=i)h_i(x)\right]\right| \le \nu, \forall h. \tag{E.40}$$

Equivalently, this can be re-written in terms of the conditional class probabilities $\eta^{\mathrm{tr}}(x) = \mathbf{P}^{\mu}(y=1|x)$:

$$\left|\mathcal{E}^{\mathrm{lin}}[h] \,-\, \mathbf{E}_{x\sim\mathbf{P}^{\mu}}\left[\sum_{i=1}^{k}\bar{W}_i(x)\eta_i^{\mathrm{tr}}(x)h_i(x)\right]\right| \le \nu, \forall h, \tag{E.41}$$

where $\mathbf{P}^{\mu}$ denotes the marginal distribution of μ over $\mathcal{X}$. Denoting $h^* \in \mathrm{argmax}_h\, \mathcal{E}^{\mathrm{lin}}[h]$, we then have from (E.41),

$$\begin{aligned}
&\max_h \mathcal{E}^{\mathrm{lin}}[h] \,-\, \mathcal{E}^{\mathrm{lin}}[\widehat{h}]\\
&= \sum_{i=1}^{k}\mathbf{E}_x\left[\bar{W}_i(x)\eta_i^{\mathrm{tr}}(x)h_i^*(x))\right] \,-\, \sum_{i=1}^{k}\mathbf{E}_x\left[\bar{W}_i(x)\eta_i^{\mathrm{tr}}(x)\widehat{h}_i(x)\right] \,+\, 2\nu\\
&\le \sum_{i=1}^{k}\mathbf{E}_x\left[\widehat{W}_i(x)\eta_i^{\mathrm{tr}}(x)h_i^*(x))\right] \,-\, \sum_{i=1}^{k}\mathbf{E}_x\left[\widehat{W}_i(x)\eta_i^{\mathrm{tr}}(x)\widehat{h}_i(x)\right] \,+\, 2\nu \,+\, 2\sqrt{Lk}\kappa\\
&\qquad\qquad\qquad\qquad\text{(from (E.37), } \textstyle\sum_{i=1}^{k}\eta_i^{\mathrm{tr}}(x) = 1 \text{ and } h_i(x)\le 1\text{)}\\
&\le \sum_{i=1}^{k}\mathbf{E}_x\left[\widehat{W}_i(x)\eta_i^{\mathrm{tr}}(x)h_i^*(x))\right] \,-\, \sum_{i=1}^{k}\mathbf{E}_x\left[\widehat{W}_i(x)\widehat{\eta}_i^{\mathrm{tr}}(x)h_i^*(x))\right]\\
&\quad + \sum_{i=1}^{k}\mathbf{E}_x\left[\widehat{W}_i(x)\widehat{\eta}_i^{\mathrm{tr}}(x)h_i^*(x))\right] \,-\, \sum_{i=1}^{k}\mathbf{E}_x\left[\widehat{W}_i(x)\eta_i^{\mathrm{tr}}(x)\widehat{h}_i(x)\right] \,+\, 2\nu \,+\, 2\sqrt{Lk}\kappa
\end{aligned} \tag{E.42}$$

From definition of $\widehat{h}$ in (E.36), we have that $\sum_{i=1}^{k}\widehat{W}_i(x)\widehat{\eta}_i^{\mathrm{tr}}(x)\widehat{h}_i(x) \ge \sum_{i=1}^{k}\widehat{W}_i(x)\widehat{\eta}_i^{\mathrm{tr}}(x)h_i(x)$, for all $h:\mathcal{X}\to\Delta_k$. Therefore,

$$\begin{aligned}
&\max_h \mathcal{E}^{\mathrm{lin}}[h] \,-\, \mathcal{E}^{\mathrm{lin}}[\widehat{h}]\\
&\le \sum_{i=1}^{k}\mathbf{E}_x\left[\widehat{W}_i(x)\eta_i^{\mathrm{tr}}(x)h_i^*(x))\right] \,-\, \sum_{i=1}^{k}\mathbf{E}_x\left[\widehat{W}_i(x)\widehat{\eta}_i^{\mathrm{tr}}(x)h_i^*(x))\right] \,+\, 2\nu \,+\, 2\sqrt{Lk}\kappa\\
&\quad + \sum_{i=1}^{k}\mathbf{E}_x\left[\widehat{W}_i(x)\widehat{\eta}_i^{\mathrm{tr}}(x)\widehat{h}_i(x))\right] \,-\, \sum_{i=1}^{k}\mathbf{E}_x\left[\widehat{W}_i(x)\eta_i^{\mathrm{tr}}(x)\widehat{h}_i(x)\right]
\end{aligned} \tag{E.43 cont.}$$

$$\leq \sum_{i=1}^{k} \mathbf{E}_x \left[\widehat{W}_i(x)|\eta_i^{\text{tr}}(x) - \widehat{\eta}_i^{\text{tr}}(x)||h_i^*(x) - \widehat{h}_i(x)|\right] \,+\, 2\nu \,+\, 2\sqrt{Lk}\kappa$$

$$\leq \mathbf{E}_x \left[\max_i \left(\widehat{W}_i(x)|h_i^*(x) - \widehat{h}_i(x)|\right) \|\eta(x) - \widehat{\eta}(x)\|_1\right] \,+\, 2\nu \,+\, 2\sqrt{Lk}\kappa$$

$$\leq (B + \sqrt{Lk}\kappa)\, \mathbf{E}_x \left[\|\eta(x) - \widehat{\eta}(x)\|_1\right] \,+\, 2\nu \,+\, 2\sqrt{Lk}\kappa, \tag{E.43}$$

where the last step follows from (E.39) and $|h_i(x) - \widehat{h}_i(x)| \leq 1$. This completes the proof. The second part, where $\|\boldsymbol{\beta}\| \leq Q$, follows by applying Assumption 7.1 to normalized coefficients $\boldsymbol{\beta}/\|\boldsymbol{\beta}\|$, and scaling the associated slack ν by Q. QED.

E.2.3 Proof of Theorem 7.2

We will make a couple of minor changes to the algorithm to simplify the analysis. Firstly, instead of using the same sample S^{val} for both estimating the example weights (through call to **PI-EW** in line 9) and estimating confusion matrices $\widehat{\mathbf{C}}^{\text{val}}$ (in line 10), we split S^{val} into two halves, use one half for the first step and the other half for the second step. Using independent samples for the two steps, we will be able to derive straight-forward confidence bounds on the estimated confusion matrices in each case. In our experiments however, we find the algorithm to be effective even when a common sample is used for both steps. Secondly, we modify line 8 to include a shifted version of the metric $\widehat{\mathcal{E}}^{\text{val}}$, so that later in Appendix E.4 when we handle the case of "unknown ψ", we can avoid having to keep track of an additive constant in the gradient coefficients.

Theorem E.2 ((Restated) **Error Bound for FW-EG with known** ψ)**.** Let $\mathcal{E}^D[h] = \psi(C_{11}^D[h], \ldots, C_{kk}^D[h])$ for a *known* concave function $\psi : [0,1]^k \to \mathbb{R}_+$, which is Q-Lipschitz, and λ-smooth w.r.t. the ℓ_1-norm. Let $\widehat{\mathcal{E}}^{\text{val}}[h] = \psi(\widehat{C}_{11}^{\text{val}}[h], \ldots, \widehat{C}_{kk}^{\text{val}}[h])$. Fix $\delta \in (0,1)$. Suppose Assumption 7.1 holds with slack ν, and for any linear metric $\sum_i \beta_i C_{ii}^D[h]$ with $\|\boldsymbol{\beta}\| \leq 1$, whose associated weight coefficients is $\bar{\boldsymbol{\alpha}}$ with $\|\bar{\boldsymbol{\alpha}}\| \leq B$, w.p. $\geq 1-\delta$ over draw of S^{tr} and S_1^{val}, the weight elicitation routine in Algorithm 7.1 outputs coefficients $\widehat{\boldsymbol{\alpha}}$ with $\|\widehat{\boldsymbol{\alpha}} - \bar{\boldsymbol{\alpha}}\| \leq \kappa(\delta, n^{\text{tr}}, n^{\text{val}})$, for some function $\kappa(\cdot) > 0$. Let $B' = B + \sqrt{Lk}\,\kappa(\delta/T, n^{\text{tr}}, n^{\text{val}})$. Assume $k \leq n^{\text{val}}$. Then w.p. $\geq 1-\delta$ over draws of S^{tr} and S^{val} from D and μ resp., the classifier $\widehat{h}$ output by Algorithm E.1 after T iterations satisfies:

$$\max_h \mathcal{E}^D[h] - \mathcal{E}^D[\widehat{h}] \;\leq\; 2QB'\mathbf{E}_x\left[\|\eta^{\text{tr}}(x) - \widehat{\eta}^{\text{tr}}(x)\|_1\right] + 4Q\nu + 4Q\sqrt{Lk}\,\kappa(\delta/T, n^{\text{tr}}, n^{\text{val}})$$
$$+\,\mathcal{O}\left(\lambda k\sqrt{\frac{k\log(n^{\text{val}})\log(k) + \log(k/\delta)}{n^{\text{val}}}} + \frac{\lambda}{T}\right). \tag{E.44}$$

Algorithm E.1 : Frank-Wolfe with Elicited Gradients (FW-EG) for General Diagonal Metrics

```
1:  Input: $\widehat{\mathcal{E}}^{\text{val}}$, Basis functions $\phi^1, \ldots, \phi^L : \mathcal{X} \to [0,1]$, Pre-trained $\widehat{\eta}^{\text{tr}} : \mathcal{X} \to \Delta_k$, $S^{\text{tr}} \sim \mu$, $S^{\text{val}} \sim D$ split into two halves $S_1^{\text{val}}$ and $S_2^{\text{val}}$ of sizes $\lceil n^{\text{val}}/2 \rceil$ and $\lfloor n^{\text{val}}/2 \rfloor$ respectively, $T$, $\epsilon$
2:  Initialize classifier $h^0$ and $\mathbf{c}^0 = diag(\widehat{\mathbf{C}}^{\text{val}}[h^0])$
3:  For $t = 0$ to $T - 1$ do
4:      if $\mathcal{E}^D[h] = \psi(C_{11}^D[h], \ldots, C_{kk}^D[h])$ for known $\psi$:
5:          $\boldsymbol{\beta}^t = \nabla\psi(\mathbf{c}^t)$
6:          $\widehat{\mathcal{E}}^{\text{lin}}[h] = \sum_i \beta_i^t \widehat{C}_{ii}^{\text{val}}[h]$, evaluated using $S_1^{\text{val}}$
7:      else
8:          $\widehat{\mathcal{E}}^{\text{lin}}[h] = \widehat{\mathcal{E}}^{\text{val}}[h] - \widehat{\mathcal{E}}^{\text{val}}[h^t]$, evaluated using $S_1^{\text{val}}$      {small $\epsilon$ recommended}
9:      $\widehat{f} = \textbf{PI-EW}(\widehat{\mathcal{E}}^{\text{lin}}, \phi^1, ..., \phi^L, \widehat{\eta}^{\text{tr}}, S^{\text{tr}}, S_1^{\text{val}}, h^t, \epsilon)$
10:     $\widetilde{\mathbf{c}} = diag(\widehat{\mathbf{C}}^{\text{val}}[\widehat{f}])$, evaluated using $S_2^{\text{val}}$
11:     $h^{t+1} = \big(1 - \frac{2}{t+1}\big)h^t + \frac{2}{t+1}\text{onehot}(\widehat{f})$
12:     $\mathbf{c}^{t+1} = \big(1 - \frac{2}{t+1}\big)\mathbf{c}^t + \frac{2}{t+1}\widetilde{\mathbf{c}}$
13: End For
14: Output: $\widehat{h} = h^T$
```

The proof adapts techniques from [18], who show guarantees for a Frank-Wolfe based learning algorithm with a known ψ in the *absence* of distribution shift. The main proof steps are listed below:

- Prove a generalization bound for the confusion matrices $\widehat{\mathbf{C}}^{\text{val}}$ evaluated in line 10 on the validation sample (Lemma E.6)
- Establish an error bound for the call to **PI-EW** in line 9 (Lemma E.5 in previous section)
- Combine the above two results to show that the classifier $\widehat{f}$ returned in line 9 is an approximate linear maximizer needed by the Frank-Wolfe algorithm (Lemma E.7)
- Combine Lemma E.7 with a convergence guarantee for the outer Frank-Wolfe algorithm [18, 107] (using convexity of the space of confusion matrices $\mathcal{C}$) to complete the proof (Lemmas E.8–E.9).

Lemma E.6 (Generalization bound for $\mathbf{C}^D$)**.** Fix $\delta \in (0,1)$. Let $\widehat{\eta}^{\text{tr}} : \mathcal{X} \to \Delta_m$ be a fixed class probability estimator. Let $\mathcal{G} = \{h : \mathcal{X} \to [m] \,|\, h(x) \in \operatorname{argmax}_{i \in [m]} \beta_i \widehat{\eta}_i^{\text{tr}}(x) \text{ for some } \boldsymbol{\beta} \in \mathbb{R}_+^m\}$ be the set of plug-in classifiers defined with $\widehat{\eta}^{\text{tr}}$. Let

$$\bar{\mathcal{G}} = \{h(x) = \textstyle\sum_{t=1}^T u_t h_t(x) \,|\, T \in \mathbb{N}, h_1, \ldots, h_T \in \mathcal{G}, \mathbf{u} \in \Delta_T\} \tag{E.45}$$

be the set of all randomized classifiers constructed from a finite number of plug-in classifiers in $\mathcal{G}$. Assume $m \leq n^{\text{val}}$. Then with probability at least $1 - \delta$ over draw of S^{val} from D, then for $h \in \bar{\mathcal{G}}$:

$$\|\mathbf{C}^D[h] - \widehat{\mathbf{C}}^{\text{val}}[h]\|_\infty \leq \mathcal{O}\left(\sqrt{\frac{m\log(m)\log(n^{\text{val}}) + \log(m/\delta)}{n^{\text{val}}}}\right). \tag{E.46}$$

Proof. The proof follows from standard convergence based generalization arguments, where we bound the capacity of the class of plug-in classifiers $\mathcal{G}$ in terms of its Natarajan dimension [136, 137]. Applying Theorem 21 from [137], we have that the Natarajan dimension of $\mathcal{G}$ is at most $d = k\log(k)$. Applying the generalization bound in Theorem 13 in [138], along with the assumption that $k \leq n^{\text{val}}$, we have for any $i \in [k]$, with probability at least $1 - \delta$ over draw of S^{val} from D, for any $h \in \mathcal{G}$:

$$|C^D_{ii}[h] - \widehat{C}^{\text{val}}_{ii}[h]| \leq \mathcal{O}\left(\sqrt{\frac{k\log(k)\log(n^{\text{val}}) + \log(1/\delta)}{n^{\text{val}}}}\right). \tag{E.47}$$

Further note that for any randomized classifier $\bar{h}(x) = \sum_{t=1}^T u_t h_t(x) \in \bar{\mathcal{G}}$, for some $\mathbf{u} \in \Delta_T$,

$$|C^D_{ii}[\bar{h}] - \widehat{C}^{\text{val}}_{ii}[\bar{h}]| \leq \sum_{t=1}^T u_t |C^D_{ii}[h_t] - \widehat{C}^{\text{val}}_{ii}[h_t]| \leq \mathcal{O}\left(\sqrt{\frac{k\log(k)\log(n^{\text{val}}) + \log(1/\delta)}{n^{\text{val}}}}\right), \tag{E.48}$$

where the first inequality follows from linearity of expectations. Taking a union bound over all diagonal entries $i \in [k]$ completes the proof. QED.

We next show that the call to **PI-EW** in line 9 of Algorithm 7.3 computes an approximate maximizer $\widehat{f}$ for $\widehat{\mathcal{E}}^{\text{lin}}$. This is an extension of Lemma 26 in [18].

Lemma E.7 (Approximation error in linear maximizer $\widehat{f}$)**.** For each iteration t in Algorithm 7.3, denote $\bar{\mathbf{c}}^t = diag(\mathbf{C}^D[h^t])$, and $\bar{\boldsymbol{\beta}}^t = \nabla\psi(\bar{\mathbf{c}}^t)$. Suppose the assumptions in Theorem 7.2 hold. Let $B' = B + \sqrt{Lk}\,\kappa(\delta, n^{\text{tr}}, n^{\text{val}})$. Assume $k \leq n^{\text{val}}$. Then w.p. $\geq 1 - \delta$ over draw of S^{tr} and S^{val} from μ and D resp., for any $t = 1, \ldots, T$, the classifier $\widehat{f}$ returned by ***PI-EW*** in line 9 satisfies:

$$\max_h \sum_i \bar{\beta}^t_i C^D_{ii}[h] \;-\; \sum_i \bar{\beta}^t_i C^D_{ii}[\widehat{f}] \;\leq\; QB'\mathbf{E}_x\left[\|\eta^{\text{tr}}(x) - \widehat{\eta}^{\text{tr}}(x)\|_1\right] \;+\; 2Q\nu$$
$$+\, 2Q\sqrt{Lk}\,\kappa\left(\tfrac{\delta}{T}, n^{\text{tr}}, n^{\text{val}}\right) \;+\; \mathcal{O}\left(\lambda k\sqrt{\frac{k\log(k)\log(n^{\text{val}}) + \log(k/\delta)}{n^{\text{val}}}}\right). \tag{E.49}$$

Proof. The proof uses Theorem 7.1 to bound the approximation errors in the linear maximizer $\widehat{f}$ (coupled with a union bound over T iterations), and Lemma E.6 to bound the estimation errors in the confusion matrix $\mathbf{c}^t$ used to compute the gradient $\nabla\psi(\mathbf{c}^t)$.

Recall from Algorithm 7.3 that $\mathbf{c}^t = diag(\widehat{\mathbf{C}}^{\text{val}}[h^t])$ and $\boldsymbol{\beta}^t = \nabla\psi(\mathbf{c}^t)$. Note that these are approximations to the actual quantities we are interested in $\bar{\mathbf{c}}^t = diag(\mathbf{C}^D[h^t])$ and $\bar{\boldsymbol{\beta}}^t = \nabla\psi(\bar{\mathbf{c}}^t)$, both of which are evaluated using the population confusion matrix. Also, $\|\boldsymbol{\beta}\| = \|\nabla\psi(\mathbf{c}^t)\| \leq Q$ from Q-Lipschitzness of ψ.

Fix iteration t, and let $h^* \in \operatorname{argmax}_h \sum_i \bar{\beta}^t_i C^D_{ii}[h]$ for this particular iteration. Then:

$$\begin{aligned}
&\sum_i \bar{\beta}^t_i C^D_{ii}[h^*] - \sum_i \bar{\beta}^t_i C^D_{ii}[\widehat{f}] \\
&= \sum_i \bar{\beta}^t_i C^D_{ii}[h^*] - \sum_i \beta^t_i C^D_{ii}[h^*] + \sum_i \beta^t_i C^D_{ii}[h^*] - \sum_i \beta^t_i C^D_{ii}[\widehat{f}] \\
&\qquad + \sum_i \beta^t_i C^D_{ii}[\widehat{f}] - \sum_i \bar{\beta}^t_i C^D_{ii}[\widehat{f}] \\
&\leq \|\boldsymbol{\beta}^t - \bar{\boldsymbol{\beta}}^t\|_\infty \sum_i C^D_{ii}[h^*] + \sum_i \beta^t_i C^D_{ii}[h^*] - \sum_i \beta^t_i C^D_{ii}[\widehat{f}] + \|\boldsymbol{\beta}^t - \bar{\boldsymbol{\beta}}^t\|_\infty \sum_i C^D_{ii}[\widehat{f}] \\
&\leq \|\boldsymbol{\beta}^t - \bar{\boldsymbol{\beta}}^t\|_\infty(1) + \max_h \sum_i \beta^t_i C^D_{ii}[h] - \sum_i \beta^t_i C^D_{ii}[\widehat{f}] + \|\boldsymbol{\beta}^t - \bar{\boldsymbol{\beta}}^t\|_\infty(1) \quad (\textstyle\sum_{i,j} C^D_{ij}[h] = 1) \\
&= 2\|\boldsymbol{\beta}^t - \bar{\boldsymbol{\beta}}^t\|_\infty + \max_h \sum_i \beta^t_i C^D_{ii}[h] - \sum_i \beta^t_i C^D_{ii}[\widehat{f}] \\
&= 2\|\nabla\psi(\mathbf{c}^t) - \nabla\psi(\bar{\mathbf{c}}^t)\|_\infty + \max_h \sum_i \beta^t_i C^D_{ii}[h] - \sum_i \beta^t_i C^D_{ii}[\widehat{f}] \\
&\leq 2\lambda\|\mathbf{c}^t - \bar{\mathbf{c}}^t\|_1 + \max_h \sum_i \beta^t_i C^D_{ii}[h] - \sum_i \beta^t_i C^D_{ii}[\widehat{f}] \quad (\psi \text{ is } \lambda\text{-smooth w.r.t. the } \ell_1 \text{ norm}) \\
&\leq 2\lambda k\|\mathbf{c}^t - \bar{\mathbf{c}}^t\|_\infty + \max_h \sum_i \beta^t_i C^D_{ii}[h] - \sum_i \beta^t_i C^D_{ii}[\widehat{f}] \\
&\leq \mathcal{O}\left(\lambda k\sqrt{\frac{k\log(k)\log(n^{\text{val}}) + \log(k/\delta)}{n^{\text{val}}}}\right) + QB'\mathbf{E}_x\left[\|\eta^{\text{tr}}(x) - \widehat{\eta}^{\text{tr}}(x)\|_1\right] \\
&\qquad + 2Q\sqrt{Lk}\,\kappa(\delta, n^{\text{tr}}, n^{\text{val}}) + 2Q\nu,
\end{aligned} \tag{E.50}$$

where $B' = B + \sqrt{Lk}\,\kappa(\delta, n^{\text{tr}}, n^{\text{val}})$. The last step holds with probability at least $1-\delta$ over draw of S^{val} and S^{tr}, and follows from Lemma E.6 and Lemma E.5 (using $\|\boldsymbol{\beta}^t\| \leq Q$). The first bound on $\|\mathbf{c}^t - \bar{\mathbf{c}}^t\|_\infty = \|\widehat{\mathbf{C}}^{\text{val}}[h^t] - \mathbf{C}^D[h^t]\|_\infty$ holds for any randomized classifier h^t constructed from a finite number of plug-in classifiers. The second bound on the linear maximization errors holds only for a fixed t, and so we need to take a union bound over all iterations $t = 1, \ldots, T$, to complete the proof.

Note that because we use two independent samples S^{val}_1 and S^{val}_2 for the two bounds, they each hold with high probability over draws of S^{val}_1 and S^{val}_2 respectively, and hence with high probability over draw of S^{val}. QED.

Our last two lemmas restate results from [18]. The first shows convexity of the space of confusion matrices (Proposition 10 from their paper), and the second applies a result from [107] to show convergence of the classical Frank-Wolfe algorithm with approximate linear maximization steps (Theorem 16 in [18]).

Lemma E.8 (Convexity of space of confusion matrices)**.** Let $\mathcal{C} = \{\mathit{diag}(\mathbf{C}^D[h]) \,|\, h : \mathcal{X} \to \Delta_k\}$ denote the set of all confusion matrices achieved by some randomized classifier $h : \mathcal{X} \to \Delta_k$. Then $\mathcal{C}$ is convex.

Proof. For any two confusion matrices $\mathbf{C}^1, \mathbf{C}^2 \in \mathcal{C}$, there exist classifiers $h_1, h_2 : \mathcal{X} \to \Delta_k$ such that $\mathbf{c}^1 = \mathit{diag}(\mathbf{C}^D[h_1])$ and $\mathbf{c}^2 = \mathit{diag}(\mathbf{C}^D[h_2])$. We need to show that for any $u \in [0, 1]$,

$$u\mathbf{c}^1 + (1 - u)\mathbf{c}^2 \in \mathcal{C}. \tag{E.51}$$

This is true because the randomized classifier $h(x) = uh_1(x) + (1-u)h_2(x)$ yields a confusion matrix $\mathit{diag}(\mathbf{C}^D[h]) = u\,\mathit{diag}(\mathbf{C}^D[h_1]) + (1 - u)\mathit{diag}(\mathbf{C}^D[h_2]) = u\mathbf{c}^1 + (1 - u)\mathbf{c}^2 \in \mathcal{C}$. QED.

Lemma E.9 (Frank-Wolfe with approximate linear maximization [18])**.** Let the metric $\mathcal{E}^D[h] = \psi(C^D_{11}[h], \ldots, C^D_{kk}[h])$ for a concave function $\psi : [0, 1]^k \to \mathbb{R}_+$ that is λ-smooth w.r.t. the ℓ_1-norm. For each iteration t, define $\bar{\boldsymbol{\beta}}^t = \nabla\psi(\mathit{diag}(\mathbf{C}^D[h^t]))$. Suppose line 9 of Algorithm 7.3 returns a classifier $\widehat{f}$ such that $\max_h \sum_i \bar{\beta}^t_i C^D_{ii}[h] - \sum_i \bar{\beta}^t_i C^D_{ii}[\widehat{f}] \leq \Delta, \forall t \in [T]$. Then the classifier $\widehat{h}$ output by Algorithm 7.3 after T iterations satisfies:

$$\max_h \mathcal{E}^D[h] - \mathcal{E}^D[\widehat{h}] \;\leq\; 2\Delta + \frac{8\lambda}{T + 2}. \tag{E.52}$$

Proof of Theorem 7.2. The proof follows by plugging in the result from Lemma E.7 into Lemma E.9. QED.

E.3 ERROR BOUND FOR WEIGHT ELICITATION WITH FIXED PROBING CLASSIFIERS

We first state a general error bound for Algorithm 7.1 in terms of the singular values of $\boldsymbol{\Sigma}$ for any *fixed* choices for the probing classifiers. We then bound the singular values for the fixed choices in (7.16) under some specific assumptions.

Theorem E.3 (Error bound on elicited weights with fixed probing classifiers). Let $\mathcal{E}^D[h] = \sum_i \beta_i C^D_{ii}[h]$ for some (unknown) $\boldsymbol{\beta} \in \mathbb{R}^k$, and let $\widehat{\mathcal{E}}^{\text{val}}[h] = \sum_i \beta_i \widehat{C}^{\text{val}}_{ii}[h]$. Let $\bar{\boldsymbol{\alpha}}$ be the associated coefficient in Assumption 7.1 for metric $\mathcal{E}^D$. Fix $\delta \in (0, 1)$. Then for any fixed choices of the probing classifiers $h^{\ell,i}$, we have with probability $\geq 1-\delta$ over draws of S^{tr} and S^{val} from μ and D resp., the $\widehat{\boldsymbol{\alpha}}$ output by Algorithm 7.1 satisfies: $\|\widehat{\boldsymbol{\alpha}} - \bar{\boldsymbol{\alpha}}\| \leq$

$$\mathcal{O}\left(\frac{1}{\sigma_{\min}(\mathbf{\Sigma})^2}\left(Lk\sqrt{\frac{L\log(Lk/\delta)}{n^{\text{tr}}}} + \sigma_{\max}(\mathbf{\Sigma})\sqrt{\frac{Lk\log(Lk/\delta)}{n^{\text{val}}}}\right) + \frac{\nu\sqrt{Lk}}{\sigma_{\min}(\mathbf{\Sigma})}\right), \tag{E.53}$$

where $\sigma_{\min}(\mathbf{\Sigma})$ and $\sigma_{\min}(\mathbf{\Sigma})$ are respectively the smallest and largest singular values of $\mathbf{\Sigma}$.

Proof. The proof follows the same steps as Theorem 7.1, except for the bound on $\|\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\|$. Specifically, we have from (E.31):

$$\|\widehat{\boldsymbol{\alpha}} - \bar{\boldsymbol{\alpha}}\| \leq \|\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\| + \nu\sqrt{Lk}\|\mathbf{\Sigma}^{-1}\|. \tag{E.54}$$

We next bound: $\|\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\|$

$$\begin{aligned}
&\leq \|\boldsymbol{\alpha}\|\|\mathbf{\Sigma}\|\|\mathbf{\Sigma}^{-1}\|\left(\frac{\|\mathbf{\Sigma} - \widehat{\mathbf{\Sigma}}\|}{\|\mathbf{\Sigma}\|} + \frac{\|\boldsymbol{\mathcal{E}} - \widehat{\boldsymbol{\mathcal{E}}}\|}{\|\boldsymbol{\mathcal{E}}\|}\right)\\
&\leq \|\mathbf{\Sigma}^{-1}\|^2\|\boldsymbol{\mathcal{E}}\|\left(\|\mathbf{\Sigma} - \widehat{\mathbf{\Sigma}}\| + \|\mathbf{\Sigma}\|\frac{\|\boldsymbol{\mathcal{E}} - \widehat{\boldsymbol{\mathcal{E}}}\|}{\|\boldsymbol{\mathcal{E}}\|}\right) \quad (\text{from } \boldsymbol{\alpha} = \mathbf{\Sigma}^{-1}\boldsymbol{\mathcal{E}})\\
&\leq \|\mathbf{\Sigma}^{-1}\|^2\left(\|\boldsymbol{\mathcal{E}}\|\|\mathbf{\Sigma} - \widehat{\mathbf{\Sigma}}\| + \|\mathbf{\Sigma}\|\|\boldsymbol{\mathcal{E}} - \widehat{\boldsymbol{\mathcal{E}}}\|\right)\\
&\leq \|\mathbf{\Sigma}^{-1}\|^2\left(\sqrt{Lk}\|\mathbf{\Sigma} - \widehat{\mathbf{\Sigma}}\| + \|\mathbf{\Sigma}\|\|\boldsymbol{\mathcal{E}} - \widehat{\boldsymbol{\mathcal{E}}}\|\right) \quad (\text{as } \mathcal{E}^D[h] \in [0,1])\\
&\leq \mathcal{O}\left(\frac{1}{\sigma_{\min}(\mathbf{\Sigma})^2}\left(\sqrt{Lk}\sqrt{\frac{L^2k\log(Lk/\delta)}{n^{\text{tr}}}} + \sigma_{\max}(\mathbf{\Sigma})\sqrt{\frac{Lk\log(Lk/\delta)}{n^{\text{val}}}}\right)\right),
\end{aligned} \tag{E.55}$$

where the last step follows from an adaptation of Lemma E.1 (where $\mathcal{H}$ contains the Lk fixed classifiers in (7.16)) and from Lemma E.2. The last statement holds with probability at least $1-\delta$ over draws of S^{tr} and S^{val}. Substituting this bound back in (E.54) completes the proof. QED.

We next provide a bound on the singular values of $\mathbf{\Sigma}$ for a specialized setting where the the probing classifiers $h^{\ell,i}$ are set to (7.16), the basis functions ϕ^{ℓ}'s divide the data into disjoint clusters, and the base classifier $\bar{h}$ is close to having "uniform accuracies" across all the clusters and classes.

Lemma E.10. Let $h^{\ell,i}$'s be defined as in (7.16). Suppose for any x, $\phi^{\ell}(x) \in \{0,1\}$ and $\phi^{\ell}(x)\phi^{\ell'}(x) = 0, \forall \ell \neq \ell'$. Let $p_{\ell,i} = \mathbf{E}_{(x,y)\sim\mu}[\phi^{\ell}(x)\mathbf{1}(y=i)]$. Let $\bar{h}$ be such that $\kappa - \tau \leq \Phi_i^{\mu,\ell}[\bar{h}] \leq \kappa, \forall \ell, i$ and for some $\kappa < \frac{1}{k}$ and $\tau < \kappa$. Then:

$$\sigma_{\max}(\mathbf{\Sigma}) \;\leq\; L \max_{\ell,i} p_{\ell,i} \;+\; \Delta; \qquad \sigma_{\min}(\mathbf{\Sigma}) \;\geq\; \epsilon(1-k\kappa) \min_{\ell,i} p_{\ell,i} \;-\; \Delta, \tag{E.56}$$

where $\Delta = Lk\tau \max\limits_{\ell,i} p_{\ell,i}$.

Proof. We first write the matrix $\mathbf{\Sigma}$ as $\mathbf{\Sigma} = \bar{\mathbf{\Sigma}} + \mathbf{E}$, where

$$\bar{\mathbf{\Sigma}} = \begin{bmatrix} p_{1,1}\left(\epsilon + (1-\epsilon)\kappa\right) & p_{1,2}(1-\epsilon)\kappa & \dots & p_{1,k}(1-\epsilon)\kappa & p_{2,1}\kappa & \dots & p_{L,k}\kappa \\ p_{1,1}(1-\epsilon)\kappa & p_{1,2}\left(\epsilon + (1-\epsilon)\kappa\right) & \dots & p_{1,k}(1-\epsilon)\kappa & p_{2,1}\kappa & \dots & p_{L,k}\kappa \\ & & & \vdots & & & \\ p_{1,1}\kappa & p_{1,2}\kappa & \dots & p_{1,k}\kappa & p_{2,1}\kappa & \dots & p_{L,k}\left(\epsilon + (1-\epsilon)\kappa\right) \end{bmatrix}, \tag{E.57}$$

and $\mathbf{E} \in \mathbb{R}^{Lk \times Lk}$ with each $|E_{(\ell,i),(\ell',i')}| \leq \max\limits_{\ell,i} p_{\ell,i}\left(\kappa - \Phi_i^{\mu,\ell}[\bar{h}]\right) \leq \tau \max\limits_{\ell,i} p_{\ell,i}$.

The matrix $\bar{\mathbf{\Sigma}}$ can in turn be written as a product of a *symmetric* matrix $\mathbf{A} \in \mathbb{R}^{Lk \times Lk}$ and a *diagonal* matrix $\mathbf{D} \in \mathbb{R}^{Lk \times Lk}$:

$$\bar{\mathbf{\Sigma}} = \mathbf{A}\mathbf{D}, \tag{E.58}$$

where

$$\mathbf{A} = \begin{bmatrix} \epsilon + (1-\epsilon)\kappa & (1-\epsilon)\kappa & \dots & (1-\epsilon)\kappa & \kappa & \dots & \kappa \\ (1-\epsilon)\kappa & \epsilon + (1-\epsilon)\kappa & \dots & (1-\epsilon)\kappa & \kappa & \dots & \kappa \\ & & & \vdots & & & \\ (1-\epsilon)\kappa & (1-\epsilon)\kappa & \dots & \epsilon + (1-\epsilon)\kappa & \kappa & \dots & \kappa \\ & & & \vdots & & & \\ \kappa & \kappa & \dots & \kappa & \epsilon + (1-\epsilon)\kappa & \dots & (1-\epsilon)\kappa \\ & & & \vdots & & & \\ \kappa & \kappa & \dots & \kappa & (1-\epsilon)\kappa & \dots & \epsilon + (1-\epsilon)\kappa \end{bmatrix}$$

$$\mathbf{D} = diag(p_{1,1}, \dots, p_{L,k}). \tag{E.59}$$

We can then bound the largest and smallest singular values of $\mathbf{\Sigma}$ in terms of those of $\mathbf{A}$ and $\mathbf{D}$. Using Weyl's inequality (see e.g., [134]), we have

$$\sigma_{\max}(\mathbf{\Sigma}) \leq \sigma_{\max}(\bar{\mathbf{\Sigma}}) + \|\mathbf{E}\| \leq \|\mathbf{A}\|\|\mathbf{D}\| \;+\; \|\mathbf{E}\| = \sigma_{\max}(\mathbf{A})\sigma_{\max}(\mathbf{D}) + \|\mathbf{E}\|. \tag{E.60}$$

and

$$\sigma_{\min}(\mathbf{\Sigma}) \geq \sigma_{\min}(\bar{\mathbf{\Sigma}}) - \|\mathbf{E}\| = \frac{1}{\|\bar{\mathbf{\Sigma}}^{-1}\|} - \|\mathbf{E}\| \geq \frac{1}{\|\mathbf{A}^{-1}\|\|\mathbf{D}^{-1}\|} - \|\mathbf{E}\| = \sigma_{\min}(\mathbf{A})\sigma_{\min}(\mathbf{D}) - \|\mathbf{E}\|. \tag{E.61}$$

Further, we have $\|\mathbf{E}\| \leq \|\mathbf{E}\|_F \leq Lk\tau \max\limits_{\ell,i} p_{\ell,i} = \Delta$, giving us:

$$\sigma_{\max}(\boldsymbol{\Sigma}) \leq \sigma_{\max}(\mathbf{A})\sigma_{\max}(\mathbf{D}) + \Delta. \tag{E.62}$$

$$\sigma_{\min}(\boldsymbol{\Sigma}) \geq \sigma_{\min}(\mathbf{A})\sigma_{\min}(\mathbf{D}) - \Delta. \tag{E.63}$$

All that remains is to bound the singular values of $\boldsymbol{\Sigma}$ and $\mathbf{D}$. Since $\mathbf{D}$ is a diagonal matrix, it's singular values are given by its diagonal entries:

$$\sigma_{\max}(\mathbf{D}) = \max_{\ell,i}\, p_{\ell,i}; \qquad \sigma_{\min}(\mathbf{D}) = \min_{\ell,i}\, p_{\ell,i}. \tag{E.64}$$

The matrix $\mathbf{A}$ is symmetric and has a certain block structure. It's singular values are the same as the positive magnitudes of its Eigen values. We first write out it's Lk Eigen vectors:

$$\begin{array}{rlccccl}
\mathbf{x}^{1,1} & = [& \overbrace{1,-1,0,\dots,0,}^{k\text{ entries}} & \overbrace{0,\dots,0,}^{k\text{ entries}} & \dots & \overbrace{0,\dots,0}^{k\text{ entries}} &] \\
\mathbf{x}^{1,2} & = [& 1,0,-1,\dots,0, & 0,\dots,0, & \dots & 0,\dots,0 &] \\
 & & & \vdots & & & \\
\mathbf{x}^{1,k-1} & = [& 1,0,0,\dots,-1, & 0,\dots,0, & \dots & 0,\dots,0 &] \\
\mathbf{x}^{1,k} & = [& 1,\dots,1, & -1,\dots,-1, & \dots & 0,\dots,0 &] \\
\mathbf{x}^{2,1} & = [& 0,\dots,0, & 1,-1,0,\dots,0, & \dots & 0,\dots,0 &] \\
 & & & \vdots & & & \\
\mathbf{x}^{2,k-1} & = [& 0,\dots,0, & 1,0,0,\dots,-1, & \dots & 0,\dots,0 &] \\
\mathbf{x}^{2,k} & = [& -1,\dots,-1, & 1,\dots,1, & \dots & 0,\dots,0 &] \\
 & & & \vdots & & & \\
\mathbf{x}^{L,1} & = [& 0,\dots,0, & 0,\dots,0, & \dots & 1,-1,0,\dots,0 &] \\
 & & & \vdots & & & \\
\mathbf{x}^{L,k-1} & = [& 0,\dots,0, & 0,\dots,0, & \dots & 1,0,0,\dots,-1 &] \\
\mathbf{x}^{L,k} & = [& 1,\dots,1, & 1,\dots,1, & \dots & 1,\dots,1 &]
\end{array} \tag{E.65}$$

One can then verify that the Lk Eigen values of $\mathbf{A}$ are ϵ with a multiplicity of $(L-1)k$, $\epsilon(1-k\kappa)$ with a multiplicity of $k-1$ and $(L-\epsilon)k\kappa + \epsilon$ with a multiplicity of 1. Therefore:

$$\sigma_{\max}(\mathbf{A}) \leq L; \qquad \sigma_{\min}(\mathbf{A}) = \epsilon(1-k\kappa). \tag{E.66}$$

Substituting the singular (Eigen) values of $\mathbf{A}, \mathbf{D}$ into (E.62) and (E.63) completes the proof. QED.

In the above lemma, the base classifier $\bar{h}$ is assumed to have roughly uniformly low accuracies for all classes and clusters, and the closer it is to having uniform accuracies, i.e. the smaller the value of τ, the tighter are the bounds.

We have shown a bound on the singular values of $\mathbf{\Sigma}$ for a specific setting where the basis functions ϕ^{ℓ}'s divide the data into disjoint clusters. When this is not the case (e.g. with overlapping clusters (7.8), or soft clusters (7.9)), the singular values of $\mathbf{\Sigma}$ would depend on how correlated the basis functions are.

E.4 ERROR BOUND FOR FW-EG WITH UNKNOWN ψ

In this section, we provide an error bound for Algorithm E.1 for evaluation metrics of the form $\mathcal{E}^D[h] = \psi(C^D_{11}[h], \ldots, C^D_{kk}[h])$, for a smooth, but *unknown* $\psi : \mathbb{R}^k \to \mathbb{R}_+$. In this case, we do not have a closed-form expression for the gradient of ψ, but instead apply the example weight elicitation routine in Algorithm 7.1 using probing classifiers chosen from within a small neighborhood around the current iterate h^t, where ψ is effectively linear. Specifically, we invoke Algorithm 7.1 with the current iterate h^t as the base classifier and with the radius parameter ϵ set to a small value. In the error bound that we state below for this version of the algorithm, we explicitly take into account the "slack" in using a local approximation to ψ as a proxy for its gradient.

Theorem E.4 (Error Bound for Frank Wolfe with Elicited Gradients with unknown ψ). Let $\mathcal{E}^D[h] = \psi(C^D_{11}[h], \ldots, C^D_{kk}[h])$ for an *unknown* concave $\psi : [0,1]^k \to \mathbb{R}_+$, which is Q-Lipschitz, and also λ-smooth w.r.t. the ℓ_1-norm. Let $\widehat{\mathcal{E}}^{\text{val}}[h] = \psi(\widehat{C}^{\text{val}}_{11}[h], \ldots, \widehat{C}^{\text{val}}_{kk}[h])$. Fix $\delta \in (0,1)$. Suppose Assumption 7.1 holds with slack ν. Suppose for any linear metric $\sum_i \beta_i C^D_{ii}[h]$, whose associated weight coefficients in the assumption is $\bar{\boldsymbol{\alpha}}$ with $\|\bar{\boldsymbol{\alpha}}\| \leq B$, the following holds. For any $\delta \in (0,1)$, with probability $\geq 1-\delta$ over draw of S^{tr} and S^{val}, when the weight elicitation routine in Algorithm 7.1 is given an input metric $\widehat{\mathcal{E}}^{\text{val}}$ with $|\widehat{\mathcal{E}}^{\text{val}} - \sum_i \beta_i \widehat{C}^{\text{val}}_{ii}[h]| \leq \chi, \forall h$, it outputs coefficients $\widehat{\boldsymbol{\alpha}}$ such that $\|\widehat{\boldsymbol{\alpha}} - \bar{\boldsymbol{\alpha}}\| \leq \kappa(\delta, n^{\text{tr}}, n^{\text{val}}, \chi)$, for some function $\kappa(\cdot) > 0$. Let $B' = B + \sqrt{Lk}\,\kappa(\delta, n^{\text{tr}}, n^{\text{val}}, 2\lambda\epsilon^2)$. Assume $k \leq n^{\text{val}}$.
Then w.p. $\geq 1-\delta$ over draws of S^{tr} and S^{val} from D and μ respectively, the classifier $\widehat{h}$ output by Algorithm E.1 with radius parameter ϵ after T iterations satisfies:

$$\max_h \mathcal{E}^D[h] - \mathcal{E}^D[\widehat{h}] \;\leq\; 2QB'\mathbf{E}_x\left[\|\eta^{\text{tr}}(x) - \widehat{\eta}^{\text{tr}}(x)\|_1\right] + 4Q\sqrt{Lk}\,\kappa(\delta/T, n^{\text{tr}}, n^{\text{val}}, 2\lambda\epsilon^2)$$
$$+\,4Q\nu \,+\, \mathcal{O}\left(\lambda k\sqrt{\frac{k\log(n^{\text{val}})\log(k) + \log(k/\delta)}{n^{\text{val}}}} + \frac{\lambda}{T}\right). \quad \text{(E.67)}$$

One can plug-in $\kappa(\cdot)$ with e.g. the error bound we derived for Algorithm 7.1 in Theorem 7.1, suitably modified to accommodate input metrics $\widehat{\mathcal{E}}^{\text{val}}$ that may differ from the desired linear metric by at most χ. Such modifications can be easily made to Theorem 7.1 and would result in an additional term $\sqrt{Lk}\chi$ in the error bound to take into account the additional approximation errors in computing the right-hand side of the linear system in (7.13).

Before proceeding to prove Theorem E.4, we state a few useful lemmas. The following lemma shows that because $\psi(\mathbf{C})$ is λ-smooth, it is effectively linear within a small neighborhood around $\mathbf{C}$.

Lemma E.11. Suppose ψ is λ-smooth w.r.t. the ℓ_1-norm. For each iteration t of Algorithm E.1, let $\boldsymbol{v}^t = \nabla\psi(\mathbf{c}^t)$ denote the true gradient of ψ at $\mathbf{c}^t$. Then for any classifier $h^\epsilon(x) = (1-\epsilon)h^t(x) + \epsilon h(x)$,

$$\left|\widehat{\mathcal{E}}^{\text{val}}[h^\epsilon] - \widehat{\mathcal{E}}^{\text{val}}[h^t] - \sum_i v_i^t \widehat{C}_{ii}^{\text{val}}[h^\epsilon]\right| \leq 2\lambda\epsilon^2. \tag{E.68}$$

Proof. For any randomized classifier $h^\epsilon(x) = (1-\epsilon)h^t(x) + \epsilon h(x)$,

$$\begin{aligned}
\left|\widehat{\mathcal{E}}^{\text{val}}[h^\epsilon] - \widehat{\mathcal{E}}^{\text{val}}[h^t] - \sum_i v_i^t \widehat{C}_{ii}^{\text{val}}[h^\epsilon]\right| &= \left|\psi(\mathit{diag}(\widehat{\mathbf{C}}^{\text{val}}[h^\epsilon])) - \psi(\mathit{diag}(\widehat{\mathbf{C}}^{\text{val}}[h^t])) - \sum_i v_i^t \widehat{C}_{ii}^{\text{val}}[h^\epsilon]\right| \\
&\leq \frac{\lambda}{2}\|\mathit{diag}(\widehat{\mathbf{C}}^{\text{val}}[h^\epsilon]) \;-\; \mathit{diag}(\widehat{\mathbf{C}}^{\text{val}}[h^t])\|_1^2 \\
&= \frac{\lambda}{2}\|\epsilon(\mathit{diag}(\widehat{\mathbf{C}}^{\text{val}}[h]) \;-\; \mathit{diag}(\widehat{\mathbf{C}}^{\text{val}}[h^t]))\|_1^2 \\
&= \frac{\lambda}{2}\epsilon^2\|\mathit{diag}(\widehat{\mathbf{C}}^{\text{val}}[h]) \;-\; \mathit{diag}(\widehat{\mathbf{C}}^{\text{val}}[h^t])\|_1^2 \\
&\leq \frac{\lambda}{2}\epsilon^2\left(\|\mathit{diag}(\widehat{\mathbf{C}}^{\text{val}}[h])\|_1 \;+\; \|\mathit{diag}(\widehat{\mathbf{C}}^{\text{val}}[h^t])\|_1\right)^2 \\
&\leq \frac{\lambda}{2}\epsilon^2(2)^2 \;=\; 2\lambda\epsilon^2.
\end{aligned} \tag{E.69}$$

Here the second line follows from the fact that ψ is λ-smooth w.r.t. the ℓ_1-norm,and $\boldsymbol{v}^t = \nabla\psi(\mathit{diag}(\widehat{\mathbf{C}}^{\text{val}}[h^t]))$. The third line follows from linearity of expectations. The last line follows from the fact that the sum of the entries of a confusion matrix (and hence the sum of its diagonal entries) cannot exceed 1. QED.

We next restate the error bounds for the call to **PI-EW** in line 9 and the corresponding bound on the approximation error in the linear maximizer $\widehat{f}$ obtained.

Lemma E.12 (Error bound for call to **PI-EW** in line 9 with unknown ψ)**.** For each iteration t of Algorithm 7.3, let $\boldsymbol{\upsilon}^t = \nabla\psi(\mathbf{c}^t)$ denote the true gradient of ψ at $\mathbf{c}^t$, when the algorithm is run with an unknown ψ that is Q-Lipschitz and λ-smooth w.r.t. the ℓ_1-norm. Let $\bar{\boldsymbol{\alpha}}$ be the associated weighting coefficient for the linear metric $\sum_i \upsilon_i^t C_{ii}^D[h]$ (whose coefficients are unknown) in Assumption 7.1, with $\|\bar{\boldsymbol{\alpha}}\|_1 \leq B$, and with slack ν. Fix $\delta > 0$. Suppose w.p. $\geq 1-\delta$ over draw of S^{tr} and S^{val}, when the weight elicitation routine used in ***PI-EW*** is called with the input metric $\widehat{\mathcal{E}}^{\text{val}}[h] - \widehat{\mathcal{E}}^{\text{val}}[h^t]$ with $|\widehat{\mathcal{E}}^{\text{val}}[h] - \sum_i \upsilon_i \widehat{C}_{ii}^{\text{val}}[h]| \leq \chi, \forall h$, it outputs coefficients $\widehat{\boldsymbol{\alpha}}$ such that $\|\widehat{\boldsymbol{\alpha}} - \bar{\boldsymbol{\alpha}}\| \leq \kappa(\delta, n^{\text{tr}}, n^{\text{val}}, \chi)$, for some function $\kappa(\cdot) > 0$. Let $B' = B + \sqrt{Lk}\,\kappa(\delta, n^{\text{tr}}, n^{\text{val}}, 2\lambda\epsilon^2)$. Then with the same probability, the classifier $\widehat{h}$ output by ***PI-EW*** when called by Algorithm E.1 with metric $\widehat{\mathcal{E}}^{\text{lin}}[h] = \widehat{\mathcal{E}}^{\text{val}}[h] - \widehat{\mathcal{E}}^{\text{val}}[h^t]$ and radius ϵ satisfies:

$$\max_h \sum_i \upsilon_i^t C_{ii}^D[h] - \sum_i \upsilon_i^t C_{ii}^D[\widehat{h}] \leq Q\Big(B'\mathbf{E}_x\left[\|\eta^{\text{tr}}(x) - \widehat{\eta}^{\text{tr}}(x)\|_1\right] \\ + 2\sqrt{Lk}\,\kappa(\delta, n^{\text{tr}}, n^{\text{val}}, 2\lambda\epsilon^2) \,+\, 2\nu\Big), \tag{E.70}$$

where $\eta_i^{\text{tr}}(x) = \mathbf{P}^{\mu}(y = i|x)$.

Proof. The proof is the same as that of Lemma E.5 for the "known ψ" case, except that the $\kappa(\cdot)$ guarantee for the call to weight elicitation routine in line 2 is different, and takes into account the fact that the input metric $\widehat{\mathcal{E}}^{\text{val}}[h] - \widehat{\mathcal{E}}^{\text{val}}[h^t]$ to the weight elicitation routine is only a local approximation to the (unknown) linear metric $\sum_i \upsilon_i \widehat{C}_{ii}^{\text{val}}[h]$. We use Lemma E.11 to compute the value of slack χ in $\kappa(\cdot)$. QED.

Lemma E.13 (Approximation error in linear maximizer $\widehat{f}$ in line 9 with unknown ψ)**.** For each iteration t in Algorithm E.1, let $\bar{\mathbf{c}}^t = \mathit{diag}(\mathbf{C}^D[h^t])$ and let $\bar{\boldsymbol{\beta}}^t = \nabla\psi(\bar{\mathbf{c}}^t)$ denote the unknown gradient of ψ evaluated at $\bar{\mathbf{c}}^t$. Suppose the assumptions in Theorem E.4 hold. Let $B' = B + \sqrt{Lk}\,\kappa(\delta, n^{\text{tr}}, n^{\text{val}}, 2\lambda\epsilon^2)$. Assume $k \leq n^{\text{val}}$.
Then w.p. $\geq 1-\delta$ over draw of S^{tr} and S^{val} from μ and D resp., for any $t = 1, \ldots, T$, the classifier $\widehat{f}$ returned by ***PI-EW*** in line 9 satisfies:

$$\max_h \sum_i \bar{\beta}_{ii}^t C_i^D[h] \,-\, \sum_i \bar{\beta}_i^t C_{ii}^D[\widehat{f}] \,\leq\, QB'\mathbf{E}_x\left[\|\eta^{\text{tr}}(x) - \widehat{\eta}^{\text{tr}}(x)\|_1\right] \,+\, 2Q\nu \\ + 2Q\sqrt{Lk}\,\kappa\left(\tfrac{\delta}{T}, n^{\text{tr}}, n^{\text{val}}, 2\lambda\epsilon^2\right) \,+\, \mathcal{O}\left(\lambda k\sqrt{\frac{k\log(k)\log(n^{\text{val}}) + \log(k/\delta)}{n^{\text{val}}}}\right) \tag{E.71}$$

Proof. The proof is the same as that of Lemma E.7 for the "known ψ" case, with the only difference being that we use Lemma E.12 (instead of Lemma E.5) to bound the linear maximization errors in equation (E.50). QED.

Proof of Theorem E.4. The proof follows from plugging Lemma E.13 into the Frank-Wolfe convergence guarantee in Lemma E.9 stated in Appendix E.2.3. QED.

E.5 RUNNING TIME OF ALGORITHM 7.3

We discuss how one iteration of FW-EG (Algorithm 7.3) compares with one iteration (epoch) of training a class-conditional probability estimate $\widehat{\eta}^{\text{tr}}(x) \approx \mathbf{P}^{\mu}(y = 1|x)$. In each iteration of FW-EG, we create Lk probing classifiers, where each probing classifier via (7.16) only requires perturbing the predictions of the base classifier $\bar{h} = h^t$ and hence requires $n^{\text{tr}} + n^{\text{val}}$ computations. After constructing the Lk probing classifiers, FW-EG solves a system of linear equations with Lk unknowns, where a naïve matrix inversion approach requires $O((Lk)^3)$ time. Notice that this can be further improved with efficient methods, e.g., using state-of-the-art linear regression solvers. Then FW-EG creates a plugin classifier and combines the predictions with the Frank-Wolfe style updates, requiring $Lk(n^{\text{tr}} + n^{\text{val}})$ computations. So, the overall time complexity for each iteration of FW-EG is $O\left(Lk(n^{\text{tr}} + n^{\text{val}}) + (Lk)^3\right)$. On the other hand, one iteration (epoch) of training $\widehat{\eta}^{\text{tr}}(x)$ requires $O(n^{\text{tr}}Hk)$ time, where H represents the total number of parameters in the underlying model architecture up to the penultimate layer. For deep networks such as ResNets (Sections 7.7.1 and 7.7.3), clearly, the run-time is dominated by the training of $\widehat{\eta}^{\text{tr}}(x)$, as long as L and k are relatively small compared to the number of parameters in the neural network. Thus our approach is reasonably faster than having to train the model for $\widehat{\eta}^{\text{tr}}$ in each iteration [91], training the model (such as ResNets) twice [94], or making multiple forward/backward passes on the training and validation set requiring three times the time for each epoch compared to training $\widehat{\eta}^{\text{tr}}$ [95].

E.6 PLUG-IN WITH COORDINATE-WISE SEARCH BASELINE

We describe the Plug-in [train-val] baseline used in Section 7.7, which constructs a classifier $\widehat{h}(x) \in \operatorname{argmax}_{i\in[k]} w_i\widehat{\eta}_i^{\text{val}}(x)$, by tuning the weights $w_i \in \mathbb{R}$ to maximize the given metric on the validation set . Note that there are k parameters to be tuned, and a naïve approach would be to use an k-dimensional grid search. Instead, we use a trick from [17] to decompose this search into an independent coordinate-wise search for each w_i. Specifically, one can estimate the relative weighting w_i/w_j between any pair of classes i, j by constructing a classifier of the form

$$h^{\zeta}(x) = \begin{cases} i & \text{if} \quad \zeta\widehat{\eta}_i^{\text{tr}}(x) > (1-\zeta)\widehat{\eta}_j^{\text{tr}}(x) \\ j & \text{otherwise} \end{cases}, \tag{E.72}$$

that predicts either class i or j based on which of these receives a higher (weighted) probability estimates, and (through a line search) finding the parameter $\zeta \in (0,1)$ for which h^{ζ} yields the highest validation metric:

$$w_i/w_j \approx \operatorname*{argmax}_{\zeta\in[0,1]} \widehat{\mathcal{E}}^{\text{val}}[h^{\zeta}]. \tag{E.73}$$

By fixing i to class k, and repeating this for classes $j \in [k-1]$, one can estimate w_j/w_k for each $j \in [k-1]$, and normalize the estimated related weights to get estimates for $w_1, \ldots, w_k$.

E.7 SOLVING CONSTRAINED SATISFACTION PROBLEM IN (7.14)

We describe some common special cases where one can easily identify classifiers $h^{\ell,i}$'s which satisfy the constraints in (7.14). We will make use of a pre-trained class probability model $\widehat{\eta}_i^{\text{tr}}(x) \approx \mathbf{P}^{\mu}(y = i|x)$, also used in Section 7.4 to construct the plug-in classifier in Algorithm 7.2. The hypothesis class $\mathcal{H}$ we consider is the set of all plug-in classifiers obtained by post-shifting $\widehat{\eta}^{\text{tr}}$.

We start with a binary classification problem ($k = 2$) with basis functions $\phi^{\ell}(x) = \mathbf{1}(g(x) = \ell)$, which divide the data points into L *disjoint* groups according to $g(x) \in [L]$. For this setting, one can show under mild assumptions on the data distribution that (7.14) does indeed have a feasible solution (using e.g. the geometric techniques used by [6] and also elaborated in the figure above). One such feasible $h^{\ell,i}$ predicts class $i \in \{0,1\}$ on all example belonging to group ℓ, and uses a thresholded of $\widehat{\eta}^{\text{tr}}$ for examples from other groups, with per-cluster thresholds. This would have the effect of maximizing the diagonal entry $\widehat{\Phi}_i^{\text{tr},\ell}[h^{\ell,i}]$ of $\widehat{\mathbf{\Sigma}}$ and the thresholds can be tuned so that the off-diagonal entries $\widehat{\Phi}_{i'}^{\text{tr},\ell'}[h^{\ell,i}], \forall(\ell',i') \neq (\ell,i)$ are small.

More specifically, for any $\ell \in [L], i \in \{0,1\}$, the classifier $h^{\ell,i}$ can be constructed as:

$$h^{\ell,i}(x) = \begin{cases} i & \text{if } g(x) = \ell \\ \mathbf{1}(\widehat{\eta}^{\text{tr}}(x) \leq \tau_{g(x)}) & \text{otherwise,} \end{cases} \tag{E.74}$$

where the thresholds $\tau_{\ell'} \in [0,1], \ell' \neq \ell$ can each be tuned independently using a line search to minimize $\max_{i'} \widehat{\Phi}_{i'}^{\text{tr},\ell'}[h^{\ell,i}]$. As long as $\widehat{\eta}^{\text{tr}}$ is a close approximation of $\mathbf{P}(y|x)$, the above procedure is guaranteed to find an approximately feasible solution for (7.14), provided one

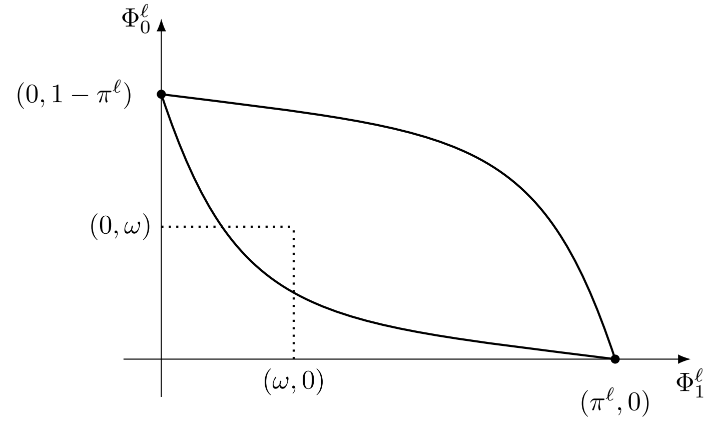


Figure E.1: Geometry of the space of Φ-confusions [6] for $k = 2$ classes and with basis functions $\phi^\ell(x) = \mathbf{1}(g(x) = \ell)$ which divide the data into L disjoint clusters. For a fixed cluster ℓ, we plot the values of $\Phi_0^{\mu,\ell}[h]$ and $\Phi_1^{\mu,\ell}[h]$ for all randomized classifiers, with $\pi^\ell = \mathbf{P}^\mu(y = 1, g(x) = \ell)$. The points on the lower boundary correspond to classifiers of the form $\mathbf{1}(\eta^{\text{tr}}(x) \leq \tau)$ for varying thresholds $\tau \in [0, 1]$. The points on the lower boundary within the dotted box correspond to the thresholded classifiers h which yield both values $\Phi_0^{\mu,\ell}[h] \leq \omega$ and $\Phi_1^{\mu,\ell}[h] \leq \omega$. One can thus find a feasible probing classifier $h^{\ell,i}$ for the constrained optimization problem in (7.14) using the construction from (E.74) as long as $\pi^\ell \geq \gamma$ and $1 - \pi^\ell \geq \gamma$, and the lower boundary intersects with the dotted box for clusters $\ell' \neq \ell$. If the latter fails, one can increase ω slowly until the classifier given in (E.74) is feasible for (7.14).

exists. Indeed one can tune the values of γ and ω in (7.14), so that the above construction (with tuned thresholds) satisfies the constraints.

We next look a multiclass problem ($k > 2$) with basis functions $\phi^\ell(x) = \mathbf{1}(g(x) = \ell)$ which again divide the data points into L *disjoint* groups. Here again, one can show under mild assumptions on the data distribution that (7.14) does indeed have a feasible solution (using e.g. the geometric tools from [17]). We can once again construct a feasible $h^{\ell,i}$ by predicting class $i \in [k]$ on all example belonging to group ℓ, and using a post-shifted classifier for examples from other groups. In particular, for any $\ell \in [L], i \in [k]$, the classifier $h^{\ell,i}$ can be constructed as:

$$h^{\ell,i}(x) = \begin{cases} i & \text{if } g(x) = \ell \\ \operatorname{argmax}_{j \in [k]} w_j^{g(x)} \widehat{\eta}_j^{\text{tr}}(x) & \text{otherwise} \end{cases}, \tag{E.75}$$

where we use k parameters $w_1^{\ell'}, \ldots, w_k^{\ell'}$ for each cluster $\ell' \neq \ell$. We can then tune these k parameters to minimize the maximum of the off-diagonal entries of $\widehat{\mathbf{\Sigma}}$, i.e. minimize $\max_{i'} \widehat{\Phi}_{i'}^{\text{tr},\ell'}[h^{\ell,i}]$. However, this may require an k-dimensional grid search. Fortunately, as described in Appendix E.6, we can use a trick from [17] to reduce the problem of tuning

Table E.1: Test macro F-measure for the maximization task in Section 6.2 of [91].

↓ Data, Method →	Adaptive Surrogates [91]	FW-EG
COMPAS	0.629	**0.652**
Adult	0.665	**0.670**
Default	0.533	0.536

k parameters into k independent line searches. This is based on the idea that the optimal relative weighting $w_i^{\ell'}/w_j^{\ell'}$ between any pair of classes can be determined through a line search. In our case, we will fix $w_k^{\ell'} = 1, \forall \ell' \neq \ell$ and compute $w_i^{\ell'}, i = 1, \ldots, k-1$ by solving the following one-dimensional optimization problem to determine the relative weighting $w_i^{\ell'}/w_k^{\ell'} = w_i^{\ell'}$.

$$w_i^{\ell'} \in \operatorname*{argmin}_{\zeta \in [0,1]} \left(\max_{i'} \widehat{\Phi}_{i'}^{\mathrm{tr},\ell'}[h^{\zeta}] \right), \quad \text{where} \;\; h^{\zeta}(x) = \begin{cases} i & \text{if} \quad \zeta \widehat{\eta}_i^{\mathrm{tr}}(x) < (1-\zeta)\widehat{\eta}_k^{\mathrm{tr}}(x) \\ k & \text{otherwise} \end{cases}. \tag{E.76}$$

We can repeat this for each cluster $\ell' \neq \ell$ to construct the (ℓ, i)-th probing classifier $h^{\ell,i}$ in (E.75).

For the more general setting, where the basis functions ϕ^{ℓ}'s cluster the data into overlapping or soft clusters (such as in (7.9)), one can find feasible classifiers for (7.14) by posing this problem as a "rate" constrained optimization problem of the form below to pick $h^{\ell,i}$:

$$\max_{h \in \mathcal{H}} \widehat{\Phi}_i^{\mathrm{tr},\ell}[h] \;\; \text{s.t.} \;\; \widehat{\Phi}_{i'}^{\mathrm{tr},\ell'}[h] \leq \omega, \forall (\ell', i') \neq (\ell, i), \tag{E.77}$$

which can be solved using off-the-shelf toolboxes such as the open-source library offered by [105].[1] Indeed one can tune the hyper-parameters γ and ω so that the solution to the above problem is feasible for (7.14). If $\mathcal{H}$ is the set of plug-in classifiers obtained by post-shifting $\widehat{\eta}^{\mathrm{tr}}$, then one can alternatively use the approach of [15] to identify the optimal post-shift on $\widehat{\eta}^{\mathrm{tr}}$ that solves the above constrained problem.

E.8 ADDITIONAL EXPERIMENTAL DETAILS

Below we provide some more details regarding the experiments:

- *Maximizing Accuracy under Label Noise on CIFAR-10 (Section 7.7.1):* The metric that we aim to optimize is test accuracy, which is a linear metric in the diagonal entries of the confusion matrix. Notice that we work with the *asymmetric* label noise model

[1] `https://github.com/google-research/tensorflow_constrained_optimization`

from Patrini et al. [94], which corresponds to the setting where a label is flipped to a particular label with a certain probability. This involves a non-diagonal noise transition matrix **T**, and consequently the corrected training objective is a linear function of the entire confusion matrix. Indeed, the loss correction approach from [94] makes use of the estimate of the entire noise-transition matrix, including the off-diagonal entries. Whereas, our approach in the experiment elicits weights for the diagonal entries alone, but assigns a different set of weights for each basis function, i.e., cluster. We are thus able to achieve better performance than [94] by optimizing correcting for the noise using a linear function of per-cluster diagonal entries. Indeed, we also observed that PI-EW often achieves better accuracy during cross-validation with ten basis functions, highlighting the benefit of underlying modeling in PI-EW. We expect to get further improvements by incorporating off-diagonal entries in PI-EW optimization on the training side as explained in Appendix E.1. We also stress that the results from our methods can be further improved by cross-validating over kernel width, UMAP dimensions, and selection of the cluster centers, which are currently set to fixed values in our experiments. Lastly, we did not compare to the Adaptive Surrogates [91] for this experiment as this baseline requires to re-train the ResNet model in every iteration, and more importantly, this method constructs its probing classifiers by perturbing the parameters of the ResNet model several times in each iteration, which can be prohibitively expensive in practice.

- *Maximizing G-mean with Proxy Labels on Adult (Section 7.7.2):* In this experiment, we use binary features as basis functions instead of RBF kernels as done in CIFAR-10 experiment. This reflects the flexibility of the proposed PI-EW and FW-EG methods. Our approach can incorporate any indicator features as basis function as long as it reflects cluster memberships. Moreover, our choice of basis function was motivated from choices made in [91]. We expect to further improve our results by incorporating more binary features as basis functions.

- *Maximizing F-measure under Domain Shift on Adience (Section 7.7.3):* As mentioned in Section 7.7.3, for the basis functions, in addition to the default basis $\phi^{\text{def}}(x) = 1\,\forall x$, we choose from subsets of six basis functions $\phi^1, \ldots, \phi^6$ that are averages of the RBFs, centered at points from the validation set corresponding to each one of the six age-gender combinations. We choose these subsets using knowledge of the underlying image classification task. Specifically, besides the default basis function, we cross-validate over three subsets of basis functions. The first subset comprises two basis functions, where the basis functions are averages of the RBF kernels with cluster centers belonging to the two true class. The second subset comprises three basis functions, where the basis functions are averages of the RBF kernels with cluster centers belonging to the three age-buckets. The third subset comprises six basis functions, where the basis functions are averages of the RBF kernels with cluster centers belonging to the combination of true class × age-bucket. We expect to further improve our results by cross-validating over kernel width and selection of the cluster centers. Lastly, we did not compare to Adaptive Surrogates, as this experiment again requires training a deep neural network model, and perturbing or retraining the model in each iteration can be prohibitively expensive in practice.

- *Maximizing Black-box Fairness Metric on Adult (Section 7.7.4):* In this experiment, since we treat the metric as a black-box, we do not assume access to gradients and thus do not run the [ψ known] variant of FW-EG. We only report the [ψ unknown] variant of FW-EG with varied basis functions as shown in Table 7.6.

- In Table E.1, we replicate the "Macro F-measure" experiment (without noise) from Section 6.2 in [91] and report results of maximizing the macro F-measure on Adult, COMPAS and Default datasets. We see that our approach yields notable gains on two out of the three datasets in comparison to Adaptive Surrogates approach [91].

REFERENCES

[1] H. C. Sox, *Medical decision making.* ACP Press, 1988.

[2] P. Dmitriev and X. Wu, "Measuring metrics," in *CIKM*, 2016.

[3] R. Caruana and A. Niculescu-Mizil, "Data mining in metric space: an empirical analysis of supervised learning performance criteria," in *ACM SIGKDD*, 2004, pp. 69–78.

[4] C. Ferri, J. Hernández-Orallo, and R. Modroiu, "An experimental comparison of performance measures for classification," *Pattern Recognition Letters*, vol. 30, no. 1, pp. 27–38, 2009.

[5] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[6] G. Hiranandani, S. Boodaghians, R. Mehta, and O. Koyejo, "Performance metric elicitation from pairwise classifier comparisons," in *The 22nd International Conference on Artificial Intelligence and Statistics*, 2019, pp. 371–379.

[7] B. Qian, X. Wang, F. Wang, H. Li, J. Ye, and I. Davidson, "Active learning from relative queries." in *IJCAI*, 2013, pp. 1614–1620.

[8] O. O. Koyejo, N. Natarajan, P. K. Ravikumar, and I. S. Dhillon, "Consistent binary classification with generalized performance metrics," in *NIPS*, 2014, pp. 2744–2752.

[9] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *ITCS*, 2012, pp. 214–226.

[10] A. Singla, E. Horvitz, P. Kohli, and A. Krause, "Learning to hire teams," in *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.

[11] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq, "Algorithmic decision making and the cost of fairness," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 797–806.

[12] S. Barocas, M. Hardt, and A. Narayanan, "Fairness in machine learning," *NIPS Tutorial*, 2017.

[13] J. Kleinberg, S. Mullainathan, and M. Raghavan, "Inherent trade-offs in the fair determination of risk scores," in *8th Innovations in Theoretical Computer Science Conference (ITCS 2017).* Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[14] A. Menon, H. Narasimhan, S. Agarwal, and S. Chawla, "On the statistical consistency of algorithms for binary classification under class imbalance," in *International Conference on Machine Learning*, 2013, pp. 603–611.

[15] H. Narasimhan, “Learning with complex loss functions and constraints,” in *International Conference on Artificial Intelligence and Statistics*, 2018, pp. 1646–1654.

[16] M. Hardt, E. Price, and N. Srebro, “Equality of opportunity in supervised learning,” in *Advances in neural information processing systems*, 2016, pp. 3315–3323.

[17] G. Hiranandani, S. Boodaghians, R. Mehta, and O. O. Koyejo, “Multiclass performance metric elicitation,” in *Advances in Neural Information Processing Systems*, 2019, pp. 9351–9360.

[18] H. Narasimhan, H. Ramaswamy, A. Saha, and S. Agarwal, “Consistent multiclass algorithms for complex performance measures,” in *ICML*, 2015, pp. 2398–2407.

[19] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should i trust you?: Explaining the predictions of any classifier,” in *ACM SIGKDD*. ACM, 2016, pp. 1135–1144.

[20] F. Doshi-Velez and B. Kim, “Towards A Rigorous Science of Interpretable Machine Learning,” *ArXiv e-prints:1702.08608*, 2017.

[21] G. Tamburrelli and A. Margara, “Towards automated *A/B* testing,” in *International Symposium on Search Based Software Engineering*. Springer, 2014, pp. 184–198.

[22] Y. Zhang, R. Bellamy, and K. Varshney, “Joint optimization of ai fairness and utility: A human-centered approach,” in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2020, pp. 400–406.

[23] E. Beauxis-Aussalet and L. Hardman, “Visualization of confusion matrix for non-expert users,” in *IEEE Conference on Visual Analytics Science and Technology (VAST)-Poster Proceedings*, 2014.

[24] O. O. Koyejo, N. Natarajan, P. K. Ravikumar, and I. S. Dhillon, “Consistent multilabel classification,” in *NIPS*, 2015, pp. 3321–3329.

[25] I. Steinwart, “How to compare different loss functions and their risks,” *Constructive Approximation*, vol. 26, no. 2, pp. 225–287, 2007.

[26] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[27] S. Boucheron, G. Lugosi, and P. Massart, *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

[28] W. N. Street, W. H. Wolberg, and O. L. Mangasarian, “Nuclear feature extraction for breast tumor diagnosis,” in *Biomedical Image Processing and Biomedical Visualization*, vol. 1905. International Society for Optics and Photonics, 1993, pp. 861–871.

[29] J. Dvorak and P. Savicky, “Softening splits in decision trees using simulated annealing,” in *International Conference on Adaptive and Natural Computing Algorithms*. Springer, 2007, pp. 721–729.

[30] F. Wauthier, M. Jordan, and N. Jojic, "Efficient ranking from pairwise comparisons," in *ICML*, 2013, pp. 109–117.

[31] R. Herbrich, "Large margin rank boundaries for ordinal regression," in *Advances in large margin classifiers.* The MIT Press, 2000, pp. 115–132.

[32] K. G. Jamieson and R. Nowak, "Active ranking using pairwise comparisons," in *NIPS*, 2011, pp. 2240–2248.

[33] F. Janssen and J. Furnkranz, "On meta-learning rule learning heuristics," in *ICDM.* IEEE, 2007, pp. 529–534.

[34] M. Peyrard, T. Botschen, and I. Gurevych, "Learning to score system summaries for better content selection evaluation." in *Proceedings of the Workshop on New Frontiers in Summarization*, 2017, pp. 74–84.

[35] N. Abe, B. Zadrozny, and J. Langford, "An iterative method for multi-class cost-sensitive learning," in *ACM SIGKDD.* ACM, 2004, pp. 3–11.

[36] K. G. Jamieson, R. Nowak, and B. Recht, "Query complexity of derivative-free optimization," in *Advances in Neural Information Processing Systems*, 2012, pp. 2672–2680.

[37] M. F. Duarte and Y. H. Hu, "Vehicle classification in distributed sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 826–838, 2004.

[38] J. P. Siebert, "Vehicle recognition using rule based methods," 1987.

[39] M. Kääriäinen, "Active learning in the non-realizable case," in *International Conference on Algorithmic Learning Theory.* Springer, 2006, pp. 63–77.

[40] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2002, pp. 133–142.

[41] B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2009.

[42] D. M. Kane, S. Lovett, S. Moran, and J. Zhang, "Active classification with comparison queries," in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS).* IEEE, 2017, pp. 355–366.

[43] L. Qian, J. Gao, and H. Jagadish, "Learning user preferences by adaptive pairwise comparison," *Proceedings of the VLDB Endowment*, vol. 8, no. 11, pp. 1322–1333, 2015.

[44] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, "Machine bias risk assessments in criminal sentencing," *ProPublica, May*, vol. 23, 2016.

[45] S. A. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E. P. Hamilton, and D. Roth, "A comparative study of fairness-enhancing interventions in machine learning," in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019, pp. 329–338.

[46] P. Lahoti, K. P. Gummadi, and G. Weikum, "ifair: Learning individually fair data representations for algorithmic decision making," in *2019 IEEE 35th International Conference on Data Engineering (ICDE).* IEEE, 2019, pp. 1334–1345.

[47] S. Barocas and A. D. Selbst, "Big data's disparate impact," *Calif. L. Rev.*, vol. 104, p. 671, 2016.

[48] A. Chouldechova, "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments," *Big data*, vol. 5, no. 2, pp. 153–163, 2017.

[49] R. Berk, H. Heidari, S. Jabbari, M. Kearns, and A. Roth, "Fairness in criminal justice risk assessments: The state of the art," *Sociological Methods & Research*, p. 0049124118782533, 2018.

[50] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Fairness-aware classifier with prejudice remover regularizer," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer, 2012, pp. 35–50.

[51] B. Woodworth, S. Gunasekar, M. I. Ohannessian, and N. Srebro, "Learning non-discriminatory predictors," in *Conference on Learning Theory*, 2017, pp. 1920–1953.

[52] A. K. Menon and R. C. Williamson, "The cost of fairness in binary classification," in *Conference on Fairness, Accountability and Transparency*, 2018, pp. 107–118.

[53] S. Yang and D. Q. Naiman, "Multiclass cancer classification based on gene expression comparison," *Statistical applications in genetics and molecular biology*, vol. 13, no. 4, pp. 477–496, 2014.

[54] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi, "Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 1171–1180.

[55] Y. Bechavod and K. Ligett, "Learning fair classifiers: A regularization-inspired approach," in *4th Workshop on Fairness, Accountability, and Transparency in Machine Learning (FATML)*, 2017.

[56] S. Opotow, "Affirmative action, fairness, and the scope of justice," *Journal of Social Issues*, vol. 52, no. 4, pp. 19–24, 1996.

[57] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression.* Springer, 2002.

[58] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, classifiaction," 1992.

[59] T. Joachims, "Svmlight: Support vector machine," *SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund*, vol. 19, no. 4, 1999.

[60] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in neural information processing systems*, 2017, pp. 3146–3154.

[61] H. Narasimhan, A. Cotter, and M. Gupta, "Optimizing generalized rate metrics with three players," in *Advances in Neural Information Processing Systems*, 2019, pp. 10 746–10 757.

[62] H. Valizadegan, R. Jin, R. Zhang, and J. Mao, "Learning to rank by optimizing ndcg measure," in *Advances in neural information processing systems*, 2009, pp. 1883–1891.

[63] G. S. Shieh, "A weighted kendall's tau statistic," *Statistics & probability letters*, vol. 39, no. 1, pp. 17–24, 1998.

[64] C. Ilvento, "Metric learning for individual fairness," *arXiv preprint arXiv:1906.00250*, 2019.

[65] D. Mukherjee, M. Yurochkin, M. Banerjee, and Y. Sun, "Two simple ways to learn individual fairness metric from data," in *ICML*, 2020.

[66] G. Goh, A. Cotter, M. Gupta, and M. P. Friedlander, "Satisfying real-world goals with dataset constraints," in *Advances in Neural Information Processing Systems*, 2016, pp. 2415–2423.

[67] M. B. Zafar, I. Valera, M. G. Rogriguez, and K. P. Gummadi, "Fairness constraints: Mechanisms for fair classification," in *Artificial Intelligence and Statistics*, 2017, pp. 962–970.

[68] A. Agarwal, A. Beygelzimer, M. Dudik, J. Langford, and H. Wallach, "A reductions approach to fair classification," in *International Conference on Machine Learning*, 2018, pp. 60–69.

[69] A. Noriega-Campero, M. A. Bakker, B. Garcia-Bulle, and A. Pentland, "Active fairness in algorithmic decision making," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 77–83.

[70] R. Binns, "On the apparent conflict between individual and group fairness," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 514–524.

[71] M. Kearns, S. Neel, A. Roth, and Z. S. Wu, "Preventing fairness gerrymandering: Auditing and learning for subgroup fairness," in *International Conference on Machine Learning*, 2018, pp. 2564–2572.

[72] T. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang, "Fairness without demographics in repeated loss minimization," in *International Conference on Machine Learning*, 2018, pp. 1929–1938.

[73] S. Gillen, C. Jung, M. Kearns, and A. Roth, "Online learning with an unknown fairness metric," in *Advances in neural information processing systems*, 2018, pp. 2600–2609.

[74] S. Wang, W. Guo, H. Narasimhan, A. Cotter, M. Gupta, and M. I. Jordan, "Robust optimization for fairness with noisy protected groups," 2020.

[75] A. Esuli and F. Sebastiani, "Optimizing text quantifiers for multivariate loss functions," *ACM Transactions on Knowledge Discovery and Data*, vol. 9, no. 4, p. Article 27, 2015.

[76] M. H. Stone, "The generalized weierstrass approximation theorem," *Mathematics Magazine*, vol. 21, no. 5, pp. 237–254, 1948.

[77] S. Lawrence, I. Burns, A. Back, A.-C. Tsoi, and C. Giles, "Neural network classification and prior class probabilities," in *Neural Networks: Tricks of the Trade*, ser. LNCS. Springer, 1998, pp. 1524:299–313.

[78] W. Liu and S. Chawla, "A quadratic mean based supervised learning model for managing data skewness," in *SDM*, 2011.

[79] A. Cotter, H. Narasimhan, and M. Gupta, "On making stochastic classifiers deterministic," in *NeurIPS*, 2019.

[80] P. Kar, S. Li, H. Narasimhan, S. Chawla, and F. Sebastiani, "Online optimization methods for the quantification problem," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1625–1634.

[81] B. G. Lindsay, M. Markatou, S. Ray, K. Yang, S.-C. Chen et al., "Quadratic distances on probabilities: A unified foundation," *The Annals of Statistics*, vol. 36, no. 2, pp. 983–1006, 2008.

[82] R. McGill, J. W. Tukey, and W. A. Larsen, "Variations of box plots," *The American Statistician*, vol. 32, no. 1, pp. 12–16, 1978.

[83] C. Boutilier, R. Patrascu, P. Poupart, and D. Schuurmans, "Constraint-based optimization and utility elicitation using the minimax decision criterion," *Artificial Intelligence*, vol. 170, no. 8-9, pp. 686–713, 2006.

[84] N. Benabbou, P. Perny, and P. Viappiani, "Incremental elicitation of choquet capacities for multicriteria choice, ranking and sorting problems," *Artificial Intelligence*, vol. 246, pp. 152–180, 2017.

[85] C. C. White, A. P. Sage, and S. Dozono, "A model of multiattribute decisionmaking and trade-off weight determination under uncertainty," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 2, pp. 223–229, 1984.

[86] P. Perny, P. Viappiani, and A. Boukhatem, "Incremental preference elicitation for decision making under risk with the rank-dependent utility model," in *Uncertainty in Artificial Intelligence*, 2016.

[87] U. Chajewska, D. Koller, and R. Parr, "Making rational decisions using adaptive utility elicitation," in *Aaai/Iaai*, 2000, pp. 363–369.

[88] D. Braziunas, "Decision-theoretic elicitation of generalized additive utilities," Ph.D. dissertation, 2012.

[89] P. Awasthi, A. Beutel, M. Kleindessner, J. Morganstern, and X. Wang, "Evaluating fairness of machine learning models under uncertain and incomplete information," in *FAccT*, 2021.

[90] C. Huang, S. Zhai, W. Talbott, M. B. Martin, S.-Y. Sun, C. Guestrin, and J. Susskind, "Addressing the loss-metric mismatch with adaptive loss alignment," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2891–2900.

[91] Q. Jiang, O. Adigun, H. Narasimhan, M. M. Fard, and M. Gupta, "Optimizing black-box metrics with adaptive surrogates," in *ICML*, 2020.

[92] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe, "Direct importance estimation for covariate shift adaptation," *Annals of the Institute of Statistical Mathematics*, vol. 60, no. 4, pp. 699–746, 2008.

[93] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," *Advances in neural information processing systems*, vol. 26, pp. 1196–1204, 2013.

[94] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1944–1952.

[95] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4334–4343.

[96] S. Zhao, M. M. Fard, H. Narasimhan, and M. Gupta, "Metric-optimized example weights," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7533–7542.

[97] A. Krizhevsky, G. Hinton et al., "Learning multiple layers of features from tiny images," 2009.

[98] E. Eidinger, R. Enbar, and T. Hassner, "Age and gender estimation of unfiltered faces," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2170–2179, 2014.

[99] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[100] D. Lewis, "Evaluating and optimizing autonomous text classification systems," in *SIGIR*, 1995.

[101] S. Daskalaki, I. Kopanas, and N. Avouris, "Evaluation of classifiers for an uneven class distribution problem," *Applied Artificial Intelligence*, vol. 20, pp. 381–417, 2006.

[102] J. Wang, Y. Liu, and C. Levy, "Fair classification with group-dependent label noise," *arXiv preprint arXiv:2011.00379*, 2020.

[103] A. K. Menon, B. Van Rooyen, and N. Natarajan, "Learning from binary labels with instance-dependent noise," *Machine Learning*, vol. 107, no. 8-10, pp. 1561–1595, 2018.

[104] A. Cotter, M. Gupta, and H. Narasimhan, "On making stochastic classifiers deterministic," in *Advances in Neural Information Processing Systems*, 2019.

[105] A. Cotter, H. Jiang, S. Wang, T. Narayan, S. You, K. Sridharan, and M. R. Gupta, "Optimization with non-differentiable constraints with applications to fairness, recall, churn, and other goals," *Journal of Machine Learning Research (JMLR)*, vol. 20, no. 172, pp. 1–59, 2019.

[106] F. Yang, M. Cisse, and S. Koyejo, "Fairness with overlapping groups," 2020.

[107] M. Jaggi, "Revisiting Frank-Wolfe: Projection-free sparse convex optimization," in *ICML*, 2013.

[108] N. Ye, K. M. Chai, W. S. Lee, and H. L. Chieu, "Optimizing f-measures: a tale of two approaches," in *Proceedings of the 29th International Conference on Machine Learning.* Omnipress, 2012, pp. 289–296.

[109] H. Narasimhan, R. Vaish, and S. Agarwal, "On the statistical consistency of plug-in classifiers for non-decomposable performance measures," in *Advances in Neural Information Processing Systems*, 2014, pp. 1493–1501.

[110] B. Yan, S. Koyejo, K. Zhong, and P. Ravikumar, "Binary classification with karmic, threshold-quasi-concave metrics," in *International Conference on Machine Learning.* PMLR, 2018, pp. 5531–5540.

[111] T. Joachims, "A support vector method for multivariate performance measures," in *Proceedings of the 22nd international conference on Machine learning.* ACM, 2005, pp. 377–384.

[112] P. Kar, H. Narasimhan, and P. Jain, "Online and stochastic gradient methods for non-decomposable loss functions," *arXiv preprint arXiv:1410.6776*, 2014.

[113] P. Kar, S. Li, H. Narasimhan, S. Chawla, and F. Sebastiani, "Online optimization methods for the quantification problem," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1625–1634.

[114] H. Narasimhan, P. Kar, and P. Jain, "Optimizing non-decomposable performance measures: A tale of two classes," in *International Conference on Machine Learning*. PMLR, 2015, pp. 199–208.

[115] E. Eban, M. Schain, A. Mackey, A. Gordon, R. Rifkin, and G. Elidan, "Scalable learning of non-decomposable objectives," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 832–840.

[116] G. Hiranandani, W. Vijitbenjaronk, S. Koyejo, and P. Jain, "Optimization and analysis of the pap@ k metric for recommender systems," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4260–4270.

[117] B. Frénay and M. Verleysen, "Classification in the presence of label noise: a survey," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, pp. 845–869, 2013.

[118] G. Csurka, "A comprehensive survey on domain adaptation for visual applications," *Domain adaptation in computer vision applications*, pp. 1–35, 2017.

[119] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of statistical planning and inference*, vol. 90, no. 2, pp. 227–244, 2000.

[120] T. Kanamori, S. Hido, and M. Sugiyama, "A least-squares approach to direct importance estimation," *The Journal of Machine Learning Research*, vol. 10, pp. 1391–1445, 2009.

[121] Z. Lipton, Y.-X. Wang, and A. Smola, "Detecting and correcting for label shift with black box predictors," in *International conference on machine learning*. PMLR, 2018, pp. 3122–3130.

[122] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. Smola, "Correcting sample selection bias by unlabeled data," *Advances in neural information processing systems*, vol. 19, pp. 601–608, 2006.

[123] J. Byrd and Z. Lipton, "What is the effect of importance weighting in deep learning?" in *International Conference on Machine Learning*. PMLR, 2019, pp. 872–881.

[124] T. Fang, N. Lu, G. Niu, and M. Sugiyama, "Rethinking importance weighting for deep learning under distribution shift," *arXiv preprint arXiv:2006.04662*, 2020.

[125] L. McInnes, J. Healy, N. Saul, and L. Großberger, "Umap: Uniform manifold approximation and projection," *Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.

[126] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[127] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning.* PMLR, 2017, pp. 1321–1330.

[128] H. Shen, H. Jin, Á. A. Cabrera, A. Perer, H. Zhu, and J. I. Hong, "Designing alternative representations of confusion matrices to support non-expert public understanding of algorithm performance," *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, no. CSCW2, pp. 1–22, 2020.

[129] R. Mazza, *Introduction to information visualization.* Springer Science & Business Media, 2009.

[130] T. H. Cormen, *Introduction to algorithms.* MIT press, 2009.

[131] S. K. Tavker, H. G. Ramaswamy, and H. Narasimhan, "Consistent plug-in classifiers for complex objectives and constraints," in *Advances in Neural Information Processing Systems*, 2020.

[132] A. Daniely, S. Sabato, S. Ben-David, and S. Shalev-Shwartz, "Multiclass learnability and the ERM principle," *JMLR*, vol. 16, no. 1, p. 2377–2404, Jan. 2015.

[133] B. K. Natarajan, "On learning sets and functions," *Machine Learning*, vol. 4, no. 1, pp. 67–97, 1989.

[134] G. W. Stewart, "Perturbation theory for the singular value decomposition," Tech. Rep., 1998.

[135] J. W. Demmel, *Applied numerical linear algebra.* SIAM, 1997.

[136] B. K. Natarajan, "On learning sets and functions," *Machine Learning*, vol. 4, no. 1, pp. 67–97, 1989.

[137] A. Daniely, S. Sabato, S. Ben-David, and S. Shalev-Shwartz, "Multiclass learnability and the erm principle," in *Proceedings of the 24th Annual Conference on Learning Theory.* JMLR Workshop and Conference Proceedings, 2011, pp. 207–232.

[138] A. Daniely, S. Sabato, S. Ben-David, and S. Shalev-Shwartz, "Multiclass learnability and the erm principle," *Journal of Machine Learning Research*, vol. 16, pp. 2377–2404, 2015.