Journal of Machine Learning Research 25 (2024) 1-81 Submitted 10/22; Published 11/24

Consistent Multiclass Algorithms for Complex Metrics and Constraints

Harikrishna Narasimhan HNARASIMHAN@GOOGLE.COM
Google Research, Mountain View, USA

Harish G. Ramaswamy HARIGURU@DSAI.IITM.AC.IN
Indian Institute of Technology Madras, Chennai, India

Shiv Kumar Tavker* TAVKER@AMAZON.COM
Amazon Inc., Bengaluru, India

Drona Khurana* DRONAKHURANA1294@GMAIL.COM
University of Colorado Boulder, USA

Praneeth Netrapalli PNETRAPALLI@GOOGLE.COM
Google Research India, Bengaluru, India

Shivani Agarwal ASHIVANI@SEAS.UPENN.EDU
University of Pennsylvania, Philadelphia, USA

Editor: Gabor Lugosi

Abstract

We present consistent algorithms for multiclass learning with complex performance metrics and constraints, where the objective and constraints are defined by arbitrary functions of the confusion matrix. This setting includes many common performance metrics such as the multiclass G-mean and micro F_1-measure, and constraints such as those on the classifier's precision and recall and more recent measures of fairness discrepancy. We give a general framework for designing consistent algorithms for such complex design goals by viewing the learning problem as an optimization problem over the set of feasible confusion matrices. We provide multiple instantiations of our framework under different assumptions on the performance metrics and constraints, and in each case show rates of convergence to the optimal (feasible) classifier (and thus asymptotic consistency). Experiments on a variety of multiclass classification tasks and fairness constrained problems show that our algorithms compare favorably to the state-of-the-art baselines.

Keywords: Multiclass, non-decomposable metrics, constraints, fairness, Frank-Wolfe, ellipsoid

1. Introduction

In many real-world machine learning tasks, the performance metric used to evaluate the performance of a classifier takes a complex form, and is not simply the expectation or sum of a loss on individual examples. Indeed, this is the case with the G-mean, H-mean and Q-mean performance metric used in class imbalance settings (Lawrence et al., 1998; Sun et al., 2006; Kennedy et al., 2009; Wang and Yao, 2012; Kim et al., 2013), the micro and macro F_1-measure used in information retrieval (IR) applications (Lewis, 1991), the worst-case error used in robust classification tasks (Vincent, 1994; Chen et al., 2017), and many others. Unlike linear performance metrics, which are simply

*Part of this work was done while SKT and DK were at the Indian Institute of Technology Madras, India.

©2024 Harikrishna Narasimhan, Harish G. Ramaswamy, Shiv Kumar Tavker, Drona Khurana, Praneeth Netrapalli and Shivani Agarwal.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v25/22-1137.html.

linear functions (defined by a loss matrix) of the confusion matrix of a classifier, these complex performance metrics are defined by general functions of the confusion matrix. In this paper, we seek to design *consistent* learning algorithms for such complex performance metrics, i.e. algorithms that converge in the limit of infinite data to the optimal classifier for the metrics.

More generally, it is common for a classifier to be evaluated on more than one performance metric, and in such cases, a desirable goal could be to optimize the classifier's performance on one metric while constraining the others to be within an acceptable range. These constrained classification problems commonly arise in fairness applications, where one may constrain a classifier to have equitable performance across multiple subgroups (Hardt et al., 2016; Zafar et al., 2017a), as well as, in many practical tasks where one wishes to constrain a classifier's precision, coverage, or churn (Eban et al., 2017; Goh et al., 2016; Cotter et al., 2019b). Such metrics and constraints can be expressed as general functions of the confusion matrix, and are categorised as complex owing to their non-decomposable structure. Standard algorithmic learning frameworks are not readily designed to handle such complexity in the objectives and constraints. Doing so requires rethinking the underlying optimization schemes, as well as conducting bespoke analysis to establish algorithmic and statistical soundness. Practical applications and the lack of general approaches to solve such problems, motivate us to address the following question:

How can we design consistent algorithms for a general learning problem where the objective and (optionally) constraints are defined by general functions of the confusion matrix?

While there has been much interest in designing consistent algorithms for various types of supervised learning problems, most of this work has focused on linear performance metrics. This includes work on the binary or multiclass 0-1 loss (Bartlett et al., 2006; Zhang, 2004a,b; Lee et al., 2004; Tewari and Bartlett, 2007), losses for specific problems such as multilabel classification (Gao and Zhou, 2011), ranking (Duchi et al., 2010; Ravikumar et al., 2011; Calauzènes et al., 2012; Yang and Koyejo, 2020), and classification with abstention (Yuan and Wegkamp, 2010; Ramaswamy et al., 2018; Finocchiaro et al., 2020), and some work on general multiclass loss matrices (Steinwart, 2007; Ramaswamy and Agarwal, 2012; Pires et al., 2013; Ramaswamy et al., 2013; Nowak-Vila et al., 2020). The design of consistent algorithms for constrained classification problems has also received much attention recently, particularly in the context of fairness (Agarwal et al., 2018; Kearns et al., 2018; Donini et al., 2018), with the focus largely being on linear metrics and constraints.

There has also been much interest in designing algorithms for more complex performance metrics. One of the seminal approaches in this area is the SVM$^{\text{perf}}$ algorithm (Joachims, 2005), which was developed primarily for the binary setting. Other examples include convex relaxation based approaches that seek to improve upon the performance of this method (Kar et al., 2014, 2016; Narasimhan et al., 2019), as well as, algorithms for the binary F_1-measure and its multiclass and multilabel variants (Dembczynski et al., 2011, 2013; Natarajan et al., 2016; Zhang et al., 2020). Parallelly, there has been increasing interest in designing *consistent* algorithms for complex performance metrics. Most of these methods are focused on the binary case (Menon et al., 2013; Koyejo et al., 2014; Narasimhan et al., 2014; Dembczyński et al., 2017), and typically require tuning a single threshold or cost parameter to optimize the metric at hand. However, this simple approach of performing a one-dimensional parameter search does not extend easily to general n-class problems, where one may need to search over as many as n^2 parameters, requiring time exponential in n^2.

In this paper, we develop a general framework for designing statistically consistent and computationally efficient algorithms for complex multiclass performance metrics and constraints. Our key

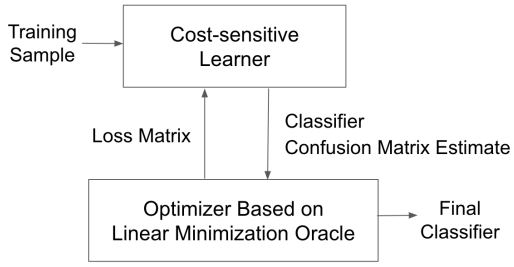


Figure 1: Simplified overview of the proposed framework.

idea is to pose the learning problem as an optimization problem over the set of feasible and achievable confusion matrices, and to solve this optimization problem using an optimization method that needs access to only a *linear minimization* routine (see Figure 1 for a simplified overview of the approach). Each of these linear minimization steps can be formulated as a *cost-sensitive learning* task, a classical problem for which numerous off-the-shelf solvers are available.

We provide instantiations of our framework under different assumptions on the performance metrics and constraints, and in each case establish rates of convergence to the optimal (feasible) classifier. Our algorithms can be used to learn plug-in type classifiers that post-shift a pre-trained class probability model, and are shown to be effective in optimizing for the given metric and constraints on a variety of application tasks.

1.1 Further Related Work

The literature on complex performance metrics and constrained learning can be broadly divided into two categories: algorithms that use surrogate relaxations (Joachims, 2005; Kar et al., 2014; Narasimhan et al., 2015a; Kar et al., 2016; Sanyal et al., 2018; Narasimhan et al., 2019), and algorithms that use a plug-in classifier (Ye et al., 2012; Menon et al., 2013; Koyejo et al., 2014; Narasimhan et al., 2014; Parambath et al., 2014; Dembczyński et al., 2017; Yang et al., 2020). The former methods are sometimes dubbed as *in-training* approaches, while the latter methods are referred to as *post-hoc* approaches.

A prominent example in the first category is the SVM$^{\text{perf}}$ method of Joachims (2005), which employs a structural SVM formulation to construct convex surrogates for complex binary evaluation metrics. This approach does not however extend to multiclass problems as it uses a cutting-plane finding routine whose running time grows exponentially with the number of classes. Moreover, follow-up work has shown that structural SVM style surrogates are not necessarily consistent for complex metrics (Dembczynski et al., 2013). More recent surrogate-based algorithms improve upon this method, offering faster training procedures and better empirical performance (Narasimhan et al., 2015a; Kar et al., 2016; Sanyal et al., 2018), but do not come with consistency guarantees.

The second category of algorithms, which construct a classifier by tuning thresholds on a class-probability estimator, do enjoy consistency guarantees, but the bulk of the work here has focused on unconstrained binary metrics (Ye et al., 2012; Menon et al., 2013; Koyejo et al., 2014; Narasimhan et al., 2014; Dembczyński et al., 2017), and for the reasons mentioned in the introduction, do not directly extend to multi-class problems.

The work that most closely relates to our paper is that of Narasimhan et al. (2019), where a family of algorithms is provided for optimizing complex metrics with and without constraints, which

includes as special cases some previous surrogate-based algorithms (Narasimhan et al., 2015a; Kar et al., 2016), as well as, the Frank-Wolfe based algorithm that appeared in a conference version of this paper (Narasimhan et al., 2015b). Their key idea is to introduce auxiliary variables to reformulate the learning task into a min-max problem, in which the minimization step entails solving a linear objective. They then propose solving the minimization step either approximately using surrogate losses, or exactly using a linear minimization oracle. They regard the use of surrogate relaxations to be more practical, and conduct all their empirical comparisons with this approach, although the guarantees they provide only show convergence to an optimal solution for the surrogate-relaxed problem. We include their surrogate-based algorithms, available as a part of the TFCO library (Cotter et al., 2019b), as baselines in our experiments.

In contrast to the methods of Narasimhan et al. (2019), our focus is on designing algorithms that are statistically consistent, and do so using linear minimization oracles (such as plug-in classifiers) that are efficient to implement. We propose various algorithms for different problem settings, and in each case, provide consistency guarantees and rates of convergence to the optimal (feasible) classifier. For one particular problem setting (discussed in Sections 4.2 and 5.2), both the metrics involved are non-smooth convex functions of the confusion matrix. The algorithms we provide for this setting are a direct adaptation of the framework presented in Narasimhan et al. (2019), but come with complete consistency analyses.

Our paper is also closely related to the growing literature on machine learning fairness, where the use of constrained optimization has become one of the dominant approaches for enforcing fairness goals. The metrics handled here are typically *linear* functions of (group-specific) confusion matrices (Hardt et al., 2016), with the approaches proposed using both surrogate relaxations (Zafar et al., 2017a,b; Goh et al., 2016; Cotter et al., 2019a,b) and linear minimization oracles (Agarwal et al., 2018; Kearns et al., 2018; Yang et al., 2020). Recently, Celis et al. (2019) extended the work of Agarwal et al. (2018) to handle more complex fairness constraints that can be written as a difference of linear-fractional metrics, but require solving a large number of linearly-constrained sub-problems, with the number of sub-problems growing exponentially with the number of groups.

Other related work includes that of Eban et al. (2017) and Kumar et al. (2021), which use surrogate approximations to solve specialized non-decomposable constrained problems, such as maximizing precision subject to recall constraints. The work of Chen et al. (2017) provides provable algorithms to minimize the maximum among multiple linear metrics using an oracle subroutine.

Its worth noting that our work is based on the *empirical utility maximization* paradigm, where an evaluation metric is viewed as a function of expected confusion statistics (Ye et al., 2012). Prior work has also considered an alternate *decision theoretic* paradigm which evaluates the metric on a finite sample S and computes an expectation of the metric over draw of S (Waegeman et al., 2014).

1.2 Contributions

The main contributions of this paper are summarized below.

- We provide a characterization of the Bayes optimal classifier for unconstrained and constrained minimization of complex multiclass metrics (see Section 3).
- We propose a unified framework for designing consistent algorithms for complex multiclass metrics and constraints given access to a linear minimization oracle, i.e., a cost-sensitive learner (see Section 4).

- For unconstrained metrics, we identify four optimization algorithms that only require access to a linear minimization oracle. These include (i) the Frank-Wolfe method for smooth convex metrics, (ii) the gradient-descent ascent algorithm and (iii) the ellipsoid method for general convex metrics, and (iv) the bisection method for ratio-of-linear metrics (see Section 4).
- For constrained learning problems, where the classifier is required to satisfy some constraints on the confusion matrix in addition to performing well on a complex metric, we provide four algorithms as counterparts to the ones mentioned above (see Section 5).
- We show that the proposed algorithms are statistically consistent when used with a plug-in based linear minimization routine (see Section 6), and also show how they can be extended to handle fairness constraints over multiple subgroups (see Section 7).
- We conduct an extensive evaluation of the proposed algorithms on benchmark multiclass, image classification, and fair classification datasets, and show that they perform comparable to or better than the state-of-the-art approaches in each case. We also provide practical guidelines on choosing an appropriate algorithm for a given setting (see Section 8).

The following is a summary of the main differences from the conference versions of this paper (Narasimhan et al., 2015b; Narasimhan, 2018; Tavker et al., 2020).

- A definitive article on the broader topic of learning with complex metrics and constraints, with improved exposition and intuitive illustrations.
- New ellipsoid-based algorithms for convex performance metrics with a linear convergence rate (*albiet* with a dependence on dimension).
- Improved bisection-based algorithm for ratio-of-linear performance metrics with a better convergence rate for handling constraints.
- An adaptation of the gradient descent-ascent algorithm from Narasimhan et al. (2019) with a complete consistency analysis.
- Convergence results presented for a general linear minimization oracle, with the plug-in method as a special case.
- New set of experiments including benchmark image classification tasks.

All proofs not provided in the main text can be found in Appendix A.

2. Preliminaries and Examples

Notations. For $n \in \mathbb{Z}_+$, we denote $[n] = \{0, \ldots, n-1\}$. For matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$, we denote $\|\mathbf{A}\|_1 = \sum_{i,j} |A_{ij}|$ and $\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{i,j} A_{i,j} B_{i,j}$. The notation $\operatorname{argmin}^*_{i \in [n]}$ will denote ties being broken in favor of the larger number. We use Δ_n to denote the $(n-1)$-dimensional probability simplex. See Table 11 in the appendix for a summary of other common symbols in the paper.

We are interested in general multiclass learning problems with instance space $\mathcal{X} \subseteq \mathbb{R}^q$ and label space $\mathcal{Y} = [n]$. Given a finite training sample $S = ((x_1, y_1), \ldots, (x_N, y_N)) \in (\mathcal{X} \times [n])^N$, the goal is to learn a multiclass classifier $h : \mathcal{X} \to [n]$, or more generally, a *randomized* multiclass classifier $h : \mathcal{X} \to \Delta_n$, which given an instance x predicts a class label in $[n]$ according to the probability distribution specified by $h(x)$. We assume examples are drawn iid from some distribution D on

$\mathcal{X} \times [n]$, and denote the marginal distribution over $\mathcal{X}$ by μ, the class-conditional distribution by $\eta_i(x) = \mathbf{P}(Y = i \,|\, X = x)$, and the class prior probabilities by $\pi_i = \mathbf{P}(Y = i)$.

2.1 Performance Metrics Based on the Confusion Matrix

We will measure the performance of a classifier in terms of its confusion matrix.

Definition 1 (Confusion matrix)**.** *The* confusion matrix*, $\mathbf{C}[h] \in [0,1]^{n \times n}$, of a randomised classifier h w.r.t. a distribution D has entries defined as*

$$C_{ij}[h] = \mathbf{P}_{(X,Y)\sim D,\, \widehat{Y} \sim h(X)}\big(Y = i,\, \widehat{Y} = j\big)\,,$$

where $\widehat{Y} \sim h(X)$ denotes a random draw of label from $h(X)$. We can get the prior class probabilities, and fractions of instances predicted as a particular class from $\mathbf{C}[h]$ by marginalisation as follows : $\sum_j C_{ij}[h] = \mathbf{P}(Y = i) := \pi_i$, and $\sum_i C_{ij}[h] = \mathbf{P}(h(X) = j)$.

We will be interested in general, complex performance metrics that can be expressed as an arbitrary function of the entries of the confusion matrix $\mathbf{C}[h]$. For any function $\psi : [0,1]^{n \times n} \to \mathbb{R}_+$, we define the performance metric of h follows:

$$\Psi[h] = \psi(\mathbf{C}[h]).$$

We adopt the convention that *lower* values of ψ correspond to *better* performance.

As the following examples show, this formulation captures both common cost-sensitive classification, which corresponds to linear functions of the entries of the confusion matrix, and more complex performance metrics such as the G-mean, micro F_1-measure, and several others.

Example 1 (Linear performance metrics)**.** *Consider a multiclass loss matrix $\mathbf{L} \in \mathbb{R}^{n \times n}$, where L_{ij} represents the cost incurred on predicting class j when the true class is i. In such cost-sensitive learning settings (Elkan, 2001), the performance of a classifier h is measured by the expected loss on a new example from D, which is a linear function of the confusion matrix $\mathbf{C}[h]$:*

$$\Psi[h] \;=\; \mathbf{E}\big[L_{Y,h(X)}\big] \;=\; \sum_{i,j} L_{ij}\, C_{ij}[h] \;=\; \psi^{\mathbf{L}}(\mathbf{C}[h])\,,$$

where $\psi^{\mathbf{L}}(\mathbf{C}) = \langle \mathbf{L}, \mathbf{C} \rangle \;\; \forall \mathbf{C} \in [0,1]^{n \times n}$. For example, for the 0-1 loss given by $L^{\text{0-1}}_{ij} = \mathbf{1}(i \neq j)$, we have $\psi^{\text{0-1}}(\mathbf{C}) = 1 - \sum_i C_{ii}$; for the balanced 0-1 loss given by $L^{\text{bal}}_{ij} = \frac{1}{n\pi_i}\mathbf{1}(i \neq j)$, we have $\psi^{\text{bal}}(\mathbf{C}) = 1 - \frac{1}{n}\sum_i \frac{1}{\pi_i} C_{ii}$; for the absolute loss used in ordinal regression, $L^{\text{ord}}_{ij} = |i - j|$, we have $\psi^{\text{ord}}(\mathbf{C}) = \sum_{i,j} |i - j| C_{ij}$.

Example 2 (Binary performance metrics)**.** *In the binary setting, the confusion matrix of a classifier contains the proportions of true negatives ($C_{00} = \text{TN}$), false positives ($C_{01} = \text{FP}$), false negatives ($C_{10} = \text{FN}$), and true positives ($C_{11} = \text{TP}$). Our framework therefore includes any binary performance metric that is expressed as a function of these quantities, including the balanced error rate metric (Menon et al., 2013) given by $\psi^{\text{BER}}(\mathbf{C}) = \frac{1}{2}\big(\frac{\text{FP}}{\pi_1} + \frac{\text{FN}}{\pi_0}\big)$, the F_β-measure given by $\psi^{F_\beta}(\mathbf{C}) = 1 - \frac{(1+\beta^2)\,\text{TP}}{(1+\beta^2)\,\text{TP} + \beta^2\,\text{FN} + \text{FP}}$ for any $\beta > 0$, all "ratio-of-linear" binary performance metrics (Koyejo et al., 2014), and more generally, all "non-decomposable" binary performance metrics (Narasimhan et al., 2014).*

Table 1: Left: examples of complex multiclass performance metrics. Right: examples of complex constraint functions. We denote $\pi_y = \mathbf{P}(Y = y)$, τ_i is the target value for class i, and $\epsilon > 0$ is a small slack. We treat the class priors π_y as constants that are known beforehand. Rows 4–6 contain fairness metrics with m protected groups, where $A(x) \in [m]$ is the protected group for instance x, $\mu_a = \mathbf{P}(A(X) = a)$, and $\mu_{a,i} = \mathbf{P}(A(X) = a, Y = i)$. Row 5 is defined for binary labels $\mathcal{Y} = \{0, 1\}$. Rows 3–6 can be equivalently written as separate constraints on individual classes (and groups), but have been conveniently expressed in terms of the maximum constraint violation.

Metric	$\psi(\mathbf{C})$
G-mean	$1 - \left(\prod_i \frac{C_{ii}}{\pi_i}\right)^{1/n}$
H-mean	$1 - n\left(\sum_i \frac{\pi_i}{C_{ii}}\right)^{-1}$
Q-mean	$\sqrt{\frac{1}{n}\sum_i \left(1 - \frac{C_{ii}}{\pi_i}\right)^2}$
Micro F_1	$1 - \frac{2\sum_{i>0} C_{ii}}{2 - \sum_i C_{1i} - \sum_i C_{i1}}$
Macro F_1	$1 - \frac{1}{n}\sum_i \frac{2C_{ii}}{\sum_j C_{ij} + \sum_j C_{ji}}$
Min-max	$\max_i \left\{1 - \frac{C_{ii}}{\pi_i}\right\}$

Constraint Function	$\phi(\mathbf{C})$
Class i Precision	$1 - \frac{C_{ii}}{\sum_j C_{ji}} - \tau_i$
Quantification	$\sum_{i=1}^n \pi_i \log\left(\frac{\pi_i}{\sum_{j=1}^n C_{ji}}\right) - \epsilon$
Coverage	$\max_{i\in[n]} \lvert\sum_j C_{ji} - \tau_i\rvert - \epsilon$
Demographic Parity	$\max_{a\in[m],\, i\in[n]} \left\lvert\frac{1}{\mu_a}\sum_j C^a_{ji} - \sum_j C_{ji}\right\rvert - \epsilon$
Equal Opportunity	$\max_{a\in[m]} \left\lvert\frac{1}{\mu_{a1}} C^a_{11} - \frac{1}{\pi_1} C_{11}\right\rvert - \epsilon$
Equalized Odds	$\max_{a\in[m],\, i,j\in[n]} \left\lvert\frac{1}{\mu_{ai}} C^a_{ij} - \frac{1}{\pi_i} C_{ij}\right\rvert - \epsilon$

Example 3 (G-mean metric)**.** *The G-mean metric is used to evaluate both binary and multiclass classifiers in settings with class imbalance (Sun et al., 2006; Wang and Yao, 2012), and is given by*

$$\psi^{\text{GM}}(\mathbf{C}) = 1 - \left(\prod_i \frac{C_{ii}}{\pi_i}\right)^{1/n}.$$

Example 4 (Micro F_1-measure)**.** *The micro F_1-measure is widely used to evaluate multiclass classifiers in information retrieval and information extraction applications (Manning et al., 2008). Many variants have been studied; we consider here the form used in the BioNLP challenge (Kim et al., 2013), which treats class 0 as a 'default' class and is effectively given by the function*[*]

$$\psi^{\text{micro}F_1}(\mathbf{C}) = 1 - \frac{2\sum_{i\neq 0} C_{ii}}{2 - \sum_i C_{0i} - \sum_i C_{i0}}.$$

In Table 1, we provide other examples of performance metrics that are given by (complex) functions of the confusion matrix, which include the macro F_1-measure (Lewis, 1991), the H-mean (Kennedy et al., 2009), the Q-mean (Lawrence et al., 1998), and the min-max metric in detection theory (Vincent, 1994) and for worst-case performance optimization (Chen et al., 2017).

We treat the class prior probabilities π_i in the definition of a performance metric as constants that are known beforehand. So when we state that ψ is a function of the entries of $\mathbf{C}$, the class prior probabilities in the definition have no dependence on the input $\mathbf{C}$. In practice, we expect that the class priors may be either estimated from data or provided by a practitioner.

[*]Another popular variant of the micro F_1 involves averaging the entries of the 'one-versus-all' binary confusion matrices for all classes, and computing the F_1 for the averaged matrix; as pointed out by Manning et al. (2008), this form of micro F_1 effectively reduces to the 0-1 classification accuracy.

2.2 Constraints Based on the Confusion Matrix

We will also be interested in machine learning goals that can be expressed as constraints on a classifier's output. Specifically, we will consider constraints that can be expressed as a general function of the classifier's confusion matrix, i.e. constraints on h of the form $\Phi_k[h] \leq 0$, $\forall k \in [K]$, where

$$\Phi_k[h] \;=\; \phi_k(\mathbf{C}[h])$$

for some $\phi_k : [0,1]^{n\times n} \to \mathbb{R}$. As shown in the following examples, this formulation includes constraints on precision, predictive coverage, fairness criteria and many others.

Example 5 (Precision)**.** *A common goal in real-world applications is to constrain the precision of a classifier for a particular class i (i.e. the number of correct predictions for class i divided by the total number of class i predictions) to be above a certain threshold τ_i. Denoting $\phi^{\text{prec-}i}(\mathbf{C}) = 1 - \frac{C_{ii}}{\sum_j C_{ji}} - \tau_i$, this constraint can be written as $\phi^{\text{prec-}i}(\mathbf{C}) \leq 0$.*

Example 6 (Coverage)**.** *A classifier's coverage for class i is the proportion of examples that are predicted as i. Prior work has looked at constraining the coverage for different classes to match a target distribution $\tau \in \Delta_n$ (Goh et al., 2016; Cotter et al., 2019b). This can be formulated as a non-positivity constraint on the maximum coverage violation, given by $\phi^{\text{cov}}(C) = \max_i |\sum_j C_{ji} - \tau_i| - \epsilon$, for a small slack $\epsilon > 0$. A variant of this constraint in the quantification literature (Esuli and Sebastiani, 2015; Gao and Sebastiani, 2015) aims to match a classifier's coverage with the class prior distribution π, with the KL-divergence between the two distributions used as the measure of discrepancy: $\phi^{\text{KLD}}(C) = \sum_{i=1}^n \pi_i \log\left(\frac{\pi_i}{\sum_{j=1}^n C_{ji}}\right) - \epsilon$.*

We next provide examples of fairness goals in machine learning that can be expressed as constraints on (group-specific) confusion matrices. In a typical fairness setup, each instance x is associated with one of m protected groups. For convenience, we will denote the protected group for a instance x by $A(x) \in [m]$.

Definition 2 (Group-specific confusion matrix)**.** *The confusion matrix of a classifier h w.r.t. a distribution D specific to group $a \in [m]$, $\mathbf{C}^a[h] \in [0,1]^{n\times n}$, has entries defined as*

$$C^a_{ij}[h] \;=\; \mathbf{P}_{(X,Y)\sim D,\, \widehat{Y}\sim h(X)}\big(Y = i,\, \widehat{Y} = j,\, A(X) = a\big),$$

where $\widehat{Y} \sim h(X)$ denotes a random draw of label from $h(X)$. We denote the fraction of instances with protected attribute a as μ_a, i.e. $P(A(X) = a) = \mu_a = \sum_{i,j} C^a_{ij}$, and the fraction of instances with protected attribute a and label i by $\mu_{a,i}$, i.e. $P(A(X) = a, Y = i) = \mu_{a,i} = \sum_j C^a_{ij}$. Clearly, the general confusion matrix can be expressed as $C_{ij} = \sum_{a\in[m]} C^a_{ij}$.

The following fairness goals are given by general functions of the m group-specific confusion matrices $\mathbf{C}^1, \ldots, \mathbf{C}^m$, and are also summarized in Table 1.

Example 7 (Demographic parity fairness)**.** *A popular fairness criterion is demographic parity, which for a problem with binary labels $\mathcal{Y} = \{0,1\}$, requires the proportion of class-1 predictions to be the same for each protected group (Hardt et al., 2016). This can be generalized to multiclass problems by requiring the proportion of prediction for each class i to be the same for each protected group. We can enforce this criterion (approximately) by defining the demographic parity violation as $\phi^{\text{DP}}(\mathbf{C}^0, \ldots, \mathbf{C}^{m-1}) = \max_{a\in[m],\, i\in[n]} \left|\frac{1}{\mu_a}\sum_j C^a_{ji} - \sum_j C_{ji}\right| - \epsilon$, where $\epsilon > 0$ is a small slack that we allow, and requiring that $\phi^{\text{DP}}(\mathbf{C}^0, \ldots, \mathbf{C}^{m-1}) \leq 0$.*

Example 8 (Equal opportunity fairness)**.** *Another popular fairness goal for problems with binary labels $\mathcal{Y} = \{0,1\}$ is the equal opportunity criterion (Zafar et al., 2017a; Hardt et al., 2016), which requires that the true positive rates be the same for examples belonging to each group. One can approximately enforce this criterion by defining the equal opportunity violation $\phi^{\text{EOpp}}(\mathbf{C}^0,\ldots,\mathbf{C}^{m-1}) = \max_{a\in[m]}\left|\frac{1}{\mu_{a1}}C^a_{11} - \frac{1}{\pi_1}C_{11}\right| - \epsilon$ with a small slack $\epsilon > 0$, and imposing the constraint $\phi^{\text{EOpp}}(\mathbf{C}^0,\ldots,\mathbf{C}^{m-1}) \leq 0$.*

Other examples of constraints that can be defined by a general function of the confusion matrix or its generalizations include the equalized odds fairness constraint (Hardt et al., 2016), constraints on classifier churn (Cormier et al., 2016; Goh et al., 2016; Cotter et al., 2019a), constraints on the performance of a classifier on multiple data distributions with varying quality (Cotter et al., 2019a), and constraints that encode performance in select portions of the ROC or precision-recall curves (Eban et al., 2017).

For ease of exposition, we will focus on metrics and constraints that are defined by a function of the overall confusion matrix $\mathbf{C}[h]$, and discuss in Section 7 how our approach can be extended to handle metrics defined by group-specific confusion matrices for fairness problems.

2.3 Learning Problems and Consistent Algorithms

One of our goals in this paper is to design learning algorithms for optimizing a performance metric of the form $\Psi[h] = \psi(\mathbf{C}[h])$:

$$\min_{h:\mathcal{X}\to\Delta_n} \Psi[h]. \tag{OP1}$$

We will also be interested in designing consistent learning algorithms for optimizing a performance measure $\Psi[h] = \psi(\mathbf{C}[h])$ subject to constraints on $\Phi_k[h] = \phi_k(\mathbf{C}[h]),\ \forall k \in [K]$:

$$\min_{h:\mathcal{X}\to\Delta_n} \Psi[h] \quad \text{s.t.} \quad \Phi_k[h] \leq 0,\ \forall k \in [K]. \tag{OP2}$$

More specifically, we wish to design algorithms that are provably *consistent* for OP1 and OP2, in that they converge in probability to the optimal performance for these problems (and when there are constraints, to zero constraint violations) as the training sample size increases.

Definition 3 (Consistent algorithm for the unconstrained problem)**.** *We define the optimal value w.r.t. D for the unconstrained problem in* OP1 *as the minimum value of the performance measure $\Psi[h]$ over all randomized classifiers h:*

$$\Psi^*_{\text{U}} = \inf_{h:\mathcal{X}\to\Delta_n} \Psi[h].$$

We say a multiclass algorithm that given a training sample S returns a classifier $h_S : \mathcal{X}\to\Delta_n$ is consistent w.r.t. *D for* OP1 *if $\forall \nu > 0$:*

$$\mathbf{P}_{S\sim D^N}\left(\Psi[h_S] - \Psi^*_{\text{U}} > \nu\right) \to 0 \quad \textit{as } N\to\infty\,.$$

For the constrained problem, we require the algorithms to additionally converge to zero constraint violations in the large sample limit.

Definition 4 (Consistent algorithm for the constrained problem)**.** *We define the optimal value for the constrained problem in* OP2 *as the minimum value of the performance measure* $\Psi[h]$ *among all randomized classifiers* h *that satisfy the* K *constraints:*

$$\Psi^*_{\mathrm{C}} = \inf_{h:\mathcal{X}\to\Delta_n,\, \Phi_k[h]\leq 0\,\forall k} \Psi[h].$$

Given a training sample S*, we say a multiclass algorithm that, returns a classifier* $h_S : \mathcal{X}\to\Delta_n$ *is* consistent w.r.t. D *for* OP2 *if* $\forall \nu > 0$*:*

$$\mathbf{P}_{S\sim D^N}\left(\Psi[h_S] \,-\, \Psi^*_{\mathrm{C}} > \nu\right) \to 0 \quad \textit{and} \quad \mathbf{P}_{S\sim D^N}\left(\forall k,\ \Phi_k[h_S] > \nu\right) \to 0 \quad \textit{as } N\to\infty\,.$$

In developing our algorithms, we will find it useful to also define the *empirical* confusion matrix of a classifier h w.r.t. sample S, denoted by $\widehat{\mathbf{C}}[h] \in [0,1]^{n\times n}$, as

$$\widehat{C}_{ij}[h] = \frac{1}{N}\sum_{\ell=1}^{N} \mathbf{1}(y_\ell = i, h(x_\ell) = j)\,.$$

3. Bayes Optimal Classifiers

As a first step towards designing consistent algorithms, we start by examining the form of Bayes optimal classifiers for OP1 and OP2. It is well known that for the simpler linear performance measures (as is the case with cost-sensitive learning problems), any classifier that picks a class that minimizes the expected loss conditioned on the instance is optimal (see e.g. Lee et al. (2004)):

Proposition 5. *Let* $\mathbf{L} \in \mathbb{R}^{n\times n}$ *be a loss matrix. Then any (deterministic) classifier* h^* *satisfying*

$$h^*(x) \in \operatorname{argmin}_{j\in[n]} \textstyle\sum_{i=1}^n \eta_i(x) L_{ij}$$

is optimal for $\psi^{\mathbf{L}}$*, i.e.* $\langle \mathbf{L}, \mathbf{C}[h^*]\rangle = \min_{h:\mathcal{X}\to\Delta_n} \langle \mathbf{L}, \mathbf{C}[h]\rangle$.

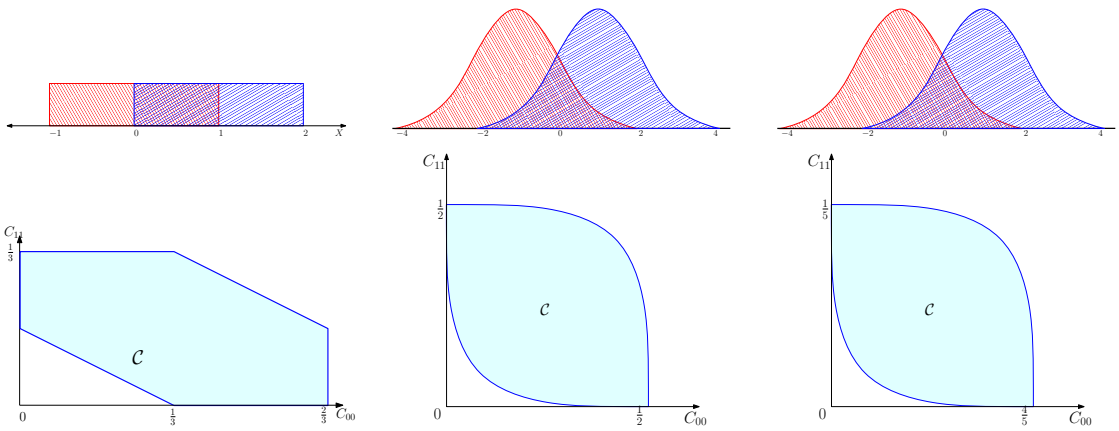
In order to understand optimal classifiers for the more complex learning problems in OP1 and OP2 described in the previous section, we will find it useful to view these learning problems as optimization problems over all *achievable confusion matrices*:

Definition 6 (Achievable confusion matrices)**.** *Define the set of* achievable confusion matrices w.r.t. D *as the set of all confusion matrices achieved by some randomized classifier:*

$$\mathcal{C} = \left\{\mathrm{vec}(\mathbf{C}[h]) \mid\ h : \mathcal{X}\to\Delta_n\right\} \subseteq \Delta_d$$

where $\mathrm{vec}(\mathbf{C}[h]) \,=\, [C_{11}[h], \ldots, C_{1n}[h], \ldots, C_{n1}[h], \ldots, C_{nn}[h]]$ *is of dimension* $d = n^2$.

See Figure 2 for an illustration of the set of achievable confusion matrices for three simple synthetic distributions, which we will refer to as `Unif`, `NormBal` and `NormImBal`. For ease of exposition, in the above definition, we represent the achievable confusion matrices by a set of flattened vectors of dimension $d = n^2$. We will also find it convenient from now on to overload notation and denote the performance measures by a function $\psi : [0,1]^d\to\mathbb{R}_+$ mapping a d-dimensional vector representation of the confusion matrix to a non-negative real number, and the constraints by functions $\phi_1, \ldots, \phi_K : [0,1]^d\to\mathbb{R}_+$ defined on d-dimensional vectors. We will similarly represent an $n \times n$ loss matrix by a flattened d-dimensional vector $\mathbf{L} \in \mathbb{R}^d$.



(a) `Unif`: $X|Y = 0$ and $X|Y = 1$ are uniform distributions over $[-1, 1]$ and $[0, 2]$ respectively; $P(Y = 1) = \frac{1}{3}$.

(b) `NormBal`: $X|Y = 0$ and $X|Y = 1$ are Gaussians with means $-\frac{1}{2}$ and $\frac{1}{2}$ respectively; $P(Y = 1) = \frac{1}{2}$.

(c) `NormImbal`: $X|Y = 0$ and $X|Y = 1$ are Gaussians with means $-\frac{1}{2}$ and $\frac{1}{2}$ respectively; $P(Y = 1) = \frac{1}{5}$.

Figure 2: The set of achievable confusion matrices $\mathcal{C}$ for three example binary-labeled distributions: (a) `Unif`, (b) `NormBal`, and (c) `NormImbal`. The top row figures show the class-conditional distributions, and the bottom row figures represent the corresponding $\mathcal{C}$. While the confusion matrix has four entries, there are only two degrees of freedom (the rows of the confusion matrix sum to the prior probabilities). We therefore only illustrate the projection of $\mathcal{C}$ on to the diagonal entries C_{00} and C_{11}. Note that the scales in the bottom row figures are different.

Proposition 7. *$\mathcal{C}$ is a convex set.*

PROOF. For any $\mathbf{C}_1, \mathbf{C}_2 \in \mathcal{C}$ and $\gamma \in [0, 1]$, we will show $\gamma\mathbf{C}_1 + (1 - \gamma)\mathbf{C}_2 \in \mathcal{C}$. Clearly, there exists randomized classifiers $h_1, h_2 : \mathcal{X} \to \Delta_n$ such that $\mathbf{C}_1 = \mathbf{C}[h_1]$ and $\mathbf{C}_2 = \mathbf{C}[h_2]$. Since $h(x) = \gamma h_1(x) + (1-\gamma)h_2(x)$ is a valid randomized classifier, $\mathbf{C}[h] = \gamma\mathbf{C}_1 + (1-\gamma)\mathbf{C}_2 \in \mathcal{C}$. □

The set $\mathcal{C}$ will play an important role in both our analysis of optimal classifiers and the subsequent development of consistent algorithms. Clearly, we can write OP1 as an unconstrained d-dimensional optimization problem over the convex set $\mathcal{C}$:

$$\min_{h:\mathcal{X}\to\Delta_n} \Psi[h] = \min_{\mathbf{C}\in\mathcal{C}} \psi(\mathbf{C}), \tag{OP1*}$$

and write OP2 as a constrained optimization problem over $\mathcal{C}$:

$$\min_{h:\mathcal{X}\to\Delta_n, \Phi_k[h]\leq 0, \forall k} \Psi[h] = \min_{\mathbf{C}\in\mathcal{C}, \boldsymbol{\phi}(\mathbf{C})\leq \mathbf{0}} \psi(\mathbf{C}), \tag{OP2*}$$

where we denote $\boldsymbol{\phi}(\mathbf{C}) = [\phi_1(\mathbf{C}), \ldots, \phi_K(\mathbf{C})]$.

3.1 Bayes Optimal Classifier for the Unconstrained Problem

While it is not clear if a classifier achieving the Bayes optimal performance exists in general, we show below that under mild assumptions, the optimal classifier for the unconstrained problem in OP1 can always be expressed as the optimal classifier for a certain linear performance metric. We

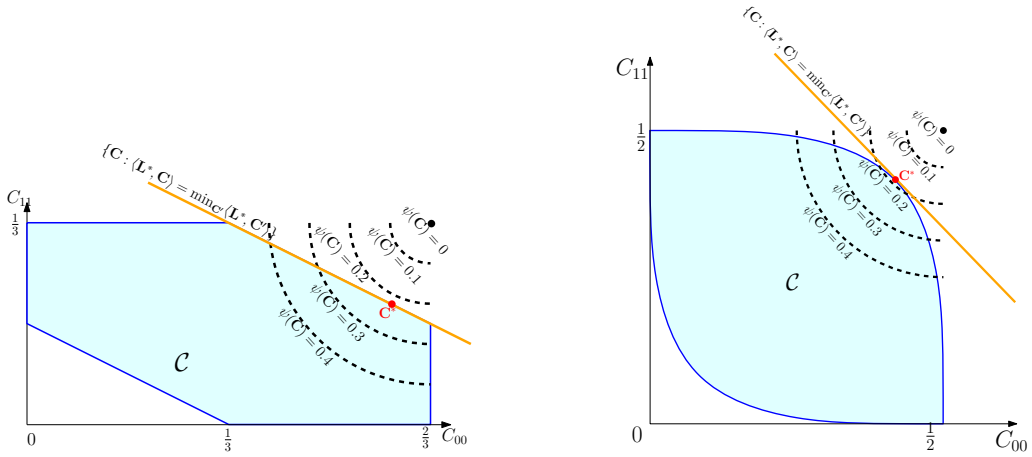


Figure 3: Illustration of the Bayes optimal classifier for the unconstrained problem in OP1 with a monotonic ψ. The figures show the set of confusion matrices $\mathcal{C}$ for distributions `Unif` (left) and `NormBal` (right) in Figure 2 (represented by the diagonal entries), the contours of the monotonic performance metric ψ, and the corresponding solution $\mathbf{C}^*$ to $\min_{\mathbf{C} \in \mathcal{C}} \psi(\mathbf{C})$ (red dot). The black dot denotes the minimizer over all confusion matrices (even those that are not achievable).

show this for "ratio-of-linear" performance measures ψ, and for "monotonic" performance measures ψ under a mild continuity assumption on D.

Proposition 8 (Bayes optimal classifier for ratio-of-linear ψ)**.** *Let the performance measure* $\psi : [0,1]^d \to \mathbb{R}_+$ *in* OP1 *be of the form* $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$ *for some* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^d$ *with* $\langle \mathbf{B}, \mathbf{C} \rangle > 0 \; \forall \mathbf{C} \in \mathcal{C}$*. Then there exists loss matrix* $\mathbf{L}^*$ *(which depends on* ψ *and* D*) such that any (deterministic) classifier that is optimal for the linear metric* $\langle \mathbf{L}^*, \mathbf{C} \rangle$ *is also optimal for* OP1*.*

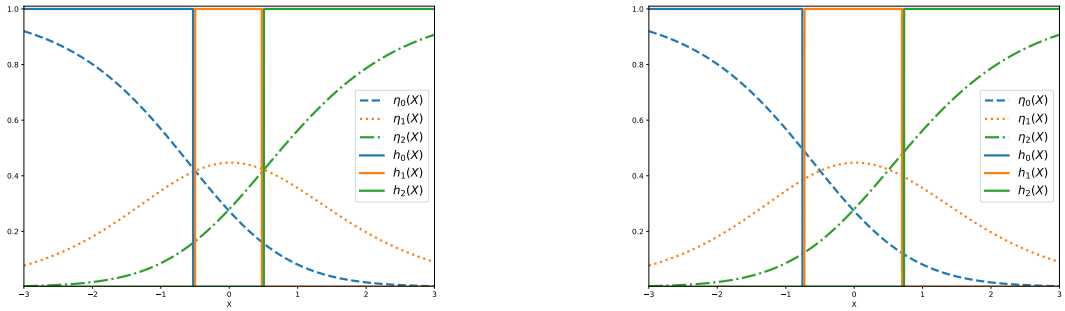
Proof. See Appendix A.1. □

Proposition 9 (Bayes optimal classifier for monotonic ψ)**.** *Let* $\psi : [0,1]^d \to \mathbb{R}_+$ *in* OP1 *be differentiable and bounded, and be strictly decreasing in* C_{ii} *for each* i *and non-decreasing in* C_{ij} *for all* $i \neq j$*. Assume* $\boldsymbol{\eta}(X)$ *is a continuous random vector. Then there exists a loss matrix* $\mathbf{L}^*$ *(which depends on* ψ *and* D*) such that any (deterministic) classifier that is optimal for the linear metric* $\langle \mathbf{L}^*, \mathbf{C} \rangle$ *over* $\mathcal{C}$ *is also optimal for* OP1*.*

Proof. See Appendix A.2. □

In Figure 3, we provide an illustration for Proposition 9 using the 2-class example distributions `Unif` and `NormBal` from Figure 2. We consider a monotonic performance metric ψ whose contours are shown overlayed in the figure with the set of feasible confusion matrices $\mathcal{C}$. It can be clearly seen that the minimal value of ψ over $\mathcal{C}$ is achieved by a point $\mathbf{C}^*$ on the boundary. Because $\mathcal{C}$ is a convex set, it follows that all points on the boundary of $\mathcal{C}$ are minimizers of some linear function $\langle \mathbf{L}, \mathbf{C} \rangle$ over $\mathbf{C} \in \mathcal{C}$. Therefore, $\mathbf{C}^*$ is also a minimizer of $\langle \mathbf{L}^*, \mathbf{C} \rangle$ for some loss matrix $\mathbf{L}^*$.

However, for $\mathbf{C}^*$ to be a unique minimizer of $\langle \mathbf{L}^*, \mathbf{C} \rangle$, we need the additional continuity assumption on $\boldsymbol{\eta}(X)$ in Proposition 9 to hold. This does not hold for the `Unif` distribution in Figure



(a) Bayes optimal classifier for 0-1 loss

(b) Bayes optimal classifier for H-mean loss

Figure 4: Comparison of Bayes optimal classifiers for the 0-1 loss (left) and the H-mean loss (right). We use a toy 3-class (denoted as class 0, 1 and 2) distribution over an one-dimensional instance space $\mathcal{X} = \mathbb{R}$, with equal priors, and with the class-conditional distribution for the three classes being a Gaussian distribution with means $-1, 0$ and 1 respectively and variance 1. We plot the conditional-class probability function $\eta_i(X)$, and the outputs of the optimal classifier $h_i^*(X)$ for each class $i \in [3]$. For the 0-1 loss, the optimal classifier predicts class 1 only on a small fraction of examples, whereas for the optimal classifier H-mean loss has greater coverage for class 1.

2a, where the corresponding conditional-class probability vectors $\boldsymbol{\eta}(X)$ take only 3 possible values in Δ_2. In contrast, $\boldsymbol{\eta}(X)$ is continuous for the `NormBal` distribution in Figure 3b, and as result, the minimizer $\mathbf{C}^*$ of $\psi(\mathbf{C})$, is also a unique minimizer for some linear function $\langle \mathbf{L}^*, \mathbf{C} \rangle$.

In Figure 4, we compare the forms of the Bayes optimal classifier for the standard 0-1 loss and for the H-mean loss in Table 1. The latter seeks to explicitly balance the classifier's performance across all classes and is a monotonic function of (the diagonal elements of) $\mathbf{C}$. We provide plots of the optimal classifiers for a toy 3-class distribution, which contains equal class priors and has a conditional-class probability distribution $\boldsymbol{\eta}(X)$ which is continuous. We know that the optimal classifier for the 0-1 loss simply outputs the label with the maximum class probability $h^*(x) = \operatorname{argmax}_i^* \eta_i(x)$. As seen in Figure 4(a), despite the class priors being equal, this classifier predicts class 1 on only a small fraction of instances. On the other hand, for the H-mean loss, Proposition 9 tells us that the optimal classifier can be obtained by minimizing some linear function of $\mathbf{C}$, the optimal classifier for which, in this particular case, is of the form $h^*(x) = \operatorname{argmax}_i^* w_i^* \eta_i(x)$, for some distribution-dependent weights $w_i^* \in \mathbb{R}_+$. Note that w_i^* can be seen as the penalty associated with a wrong prediction on class i, which in this case is the highest for class 1. The resulting classifier, shown in Figure 4(b), therefore yields equitable performance across the three classes.

3.2 Bayes Optimal Classifier for the Constrained Problem

In both the characterizations in the previous section, we show that there exists a *deterministic* classifier that is Bayes optimal for the unconstrained problem in OP1. An analogous statement does not hold in general for the constrained problem in OP2. However, we can prove a weaker characterization for OP2 showing that the Bayes optimal classifier for the problem can be expressed as a *randomized* combination of $d + 1$ deterministic classifiers.

Proposition 10 (Bayes optimal classifier for continuous $\psi, \phi_1, \ldots, \phi_K$)**.** *Let the performance measure $\psi : [0, 1]^d \to \mathbb{R}_+$ and the constraint functions $\phi_1, \ldots, \phi_K : [-1, 1]^d \to \mathbb{R}_+$ in* OP2 *be continuous*

and bounded. Then there exists $d+1$ loss matrices $\mathbf{L}_1^*, \mathbf{L}_2^*, \ldots, \mathbf{L}_{d+1}^*$ (which can depend on ψ, ϕ_k's and D) such that an optimal classifier for OP2 *can be expressed as a randomized combination of the deterministic classifiers $h_1, h_2, \ldots, h_{d+1}$, where h_i is optimal for the linear metric given by $\mathbf{L}_i^*$.*

Proof. See Appendix A.3. □

Thus for continuous and bounded $\psi, \phi_1, \ldots, \phi_K$, there *exists* a randomized classifier that minimizes OP2, and as a result a confusion matrix $\mathbf{C}^* \in \mathcal{C}$ that minimizes OP2*; this holds with no assumption on the distribution. One may also apply Proposition 10 to OP1 with $K = 0$ constraints, and show that when ψ is continuous and bounded, there exists a $\mathbf{C}^* \in \mathcal{C}$ that minimizes OP1*.

When the objective and constraints $\psi, \phi_1, \ldots, \phi_K$ together depend on fewer than $d = n^2$ entries of the confusion matrix, we can extend the above proposition to show that the number of deterministic classifiers needed to construct an optimal classifier for OP2 is at most one plus the number of confusion matrix entries the metrics depend on. For example, if we wish to optimize the G-mean metric (Example 3) subject to a constraint on the class-1 precision (Example 5), the objective and constraints together depend only on $2n - 1$ "entries" of the confusion matrix, and so an optimal classifier for this problem can be expressed as randomized combination of at most $2n$ deterministic classifiers. In Section 7, we provide a more detailed discussion about succinct vector representations for confusion matrices that require fewer than n^2 entries.

Under continuity assumptions on $\boldsymbol{\eta}(X)$ (which essentially translate to the space of achievable confusion matrices $\mathcal{C}$ being *strictly* convex), one can further show that the Bayes optimal classifier can be expressed as a randomized combination of *two* deterministic classifiers h_1 and h_2, where h_i is optimal for some linear metric $\mathbf{L}_i^*$ (Yang et al., 2020). The same characterization straight-forwardly holds for unconstrained minimization of a general performance metric ψ (Wang et al., 2019).

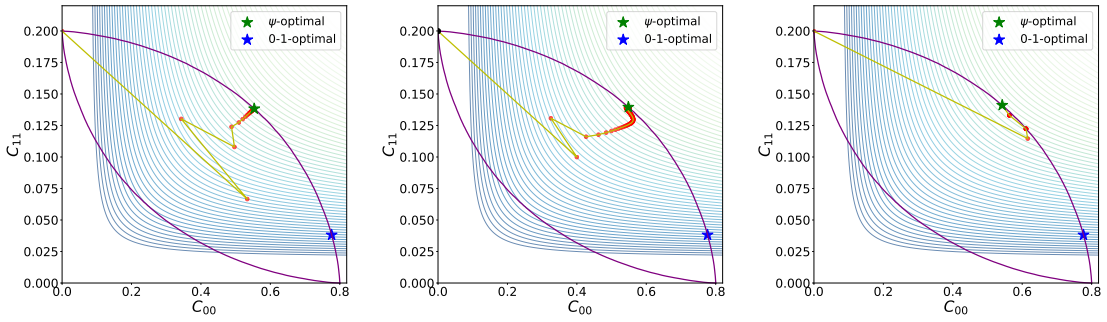
3.3 Naïve Plug-in Approach

The characterization results for the unconstrained problem in OP1 suggest a simple algorithmic approach to finding the optimal classifier: search over a large range of loss matrices $\mathbf{L}$, estimate the optimal classifier for each such $\mathbf{L}$, and select among these a classifier that yields maximal ψ-performance (e.g. on a held-out validation data set). This is the analogue of "plug-in" type methods for binary performance metrics (such as those considered by Koyejo et al. (2014) and Narasimhan et al. (2014)), where one searches over possible thresholds on the (estimated) class probability function. However, while the binary case involves a search over values for a single threshold parameter, in the multiclass case, one may need to perform a brute-force search over as many as d parameters, requiring time exponential in d. For large d, such a naïve plug-in approach is computationally intractable. In fact, this procedure becomes even more difficult to implement for the constrained problem in OP2, where the optimal classifier is a randomized combination of multiple $\mathbf{L}$-optimal classifiers, requiring a brute-force search of over multiple loss matrices $\mathbf{L}$.

In what follows, we will design efficient learning algorithms that instead search over the space of feasible confusion matrices $\mathcal{C}$ using suitable optimization methods.

4. Algorithms for Unconstrained Problems

We start with algorithms for solving the unconstrained learning problem in OP1. As a running example to illustrate our algorithms, we will use the task of maximizing the H-mean loss on the `NormImbal` distribution described in Figure 2(e).



(a) Frank-Wolfe (b) GDA (c) Ellipsoid

Figure 5: Illustration of the Frank-Wolfe (Algorithm 1), Gradient Descent-Ascent (Algorithm 2) and Ellipsoid (Algorithm 3) algorithms in minimizing the H-mean loss ψ^{HM} on the `NormImbal` distribution in Figure 2c. The figures contain the space of achievable confusion matrices $\mathcal{C}$ (with purple colored boundary), along with the contours of ψ^{HM}. The trajectory of the confusion matrix $\mathbf{C}[\bar{h}^t]$ of the averaged classifier up until iteration t is shown, where $\bar{h}^t = h^t$ for Frank-Wolfe, $\bar{h}^t = \frac{1}{t}\sum_{\tau=1}^{\tau} h^t$ for GDA, and $\bar{h}^t = \frac{1}{t}\sum_{\tau=1}^{t} \alpha_\tau^* h^\tau$ for ellipsoid, with the optimal coefficients $\boldsymbol{\alpha}^* \in \operatorname{argmin}_{\boldsymbol{\alpha}\in\Delta_t} \psi\left(\sum_{\tau=1}^{t} \alpha_\tau \mathbf{C}^\tau\right)$ computed for iterates $1, \ldots t$. The averaged classifier is seen to converge to an optimal classifier for the H-mean loss and away from that for the 0-1 loss.

Table 2: Algorithms for the unconstrained problem in OP1, with the number calls to the LMO and the optimality gap $\psi(\mathbf{C}[\bar{h}]) - \min_{\mathbf{C}\in\mathcal{C}} \psi(\mathbf{C})$ for the returned classifier $\bar{h}$. Here $\rho^{\text{eff}} = \rho + \sqrt{d}\rho'$.

Algorithm	Assumption on ψ	**# LMO Calls**	**Optimality Gap**
Frank-Wolfe	Convex, smooth, Lipschitz	$\mathcal{O}\left(1/\epsilon\right)$	$\mathcal{O}\left(\epsilon + \rho^{\text{eff}}\right)$
Gradient Descent-Ascent	Convex, Lipschitz	$\mathcal{O}\left(1/\epsilon^2\right)$	$\mathcal{O}\left(\epsilon + \rho^{\text{eff}}\right)$
Ellipsoid	Convex, Lipschitz	$\mathcal{O}\left(d^2 \log(d/\epsilon)\right)$	$\mathcal{O}\left(\epsilon + \rho^{\text{eff}}\right)$
Bisection	Ratio-of-linear	$\mathcal{O}\left(\log(1/\epsilon)\right)$	$\mathcal{O}\left(\epsilon + \rho^{\text{eff}}\right)$

As noted in our discussion of OP1*, one can view OP1 as an optimization problem over $\mathcal{C}$: $\min_{\mathbf{C}\in\mathcal{C}} \psi(\mathbf{C})$. While $\mathcal{C}$ is a convex set, it is not available directly to the learner as the set of all confusion matrices is hard to characterize. On the other hand, one operation that is easy to perform is to find an optimal classifier for a *linear* loss $\langle \mathbf{L}, \mathbf{C}\rangle$ over $\mathcal{C}$. Indeed this amounts to solving a cost-sensitive learning problem (Elkan, 2001), a task for which there are numerous classical methods available. So we assume access to an oracle for solving this linear minimization problem over $\mathcal{C}$, which takes as input a loss matrix $\mathbf{L}$ and a sample S, and outputs a classifier $\widehat{g}$ and an estimate of the confusion matrix at $\widehat{g}$ with the following properties:

Definition 11 (Linear minimization oracle)**.** *Let $\rho, \rho', \delta \in (0,1)$. A linear minimization oracle, denoted by Ω, takes a loss matrix $\mathbf{L} \in \mathbb{R}^d$ and a sample S as input, and outputs a classifier $\widehat{g}$ and a confusion matrix $\widehat{\mathbf{\Gamma}} \in \mathbb{R}^d$. We say Ω is a (ρ, ρ', δ)-approximate LMO for sample size N, if, with probability $\geq 1-\delta$ over draw of $S \sim D^N$, for any $\mathbf{L} \in \mathbb{R}^d_+$ with $\|\mathbf{L}\|_\infty \leq 1$, it outputs $(\widehat{g}, \widehat{\mathbf{\Gamma}}) = \Omega(\mathbf{L}; S)$ such that:*

$$\langle \mathbf{L}, \mathbf{C}[\widehat{g}]\rangle \leq \min_{h:\mathcal{X}\to\Delta_n} \langle \mathbf{L}, \mathbf{C}[h]\rangle + \rho; \quad \|\mathbf{C}[\widehat{g}] - \widehat{\mathbf{\Gamma}}\|_\infty \leq \rho'.$$

Algorithm 1 Frank-Wolfe (FW) Algorithm for OP1 with Smooth Convex ψ

Input: $\psi : [0,1]^d \to [0,1]$, an LMO Ω, $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, T
Initialize: $(h^0, \mathbf{C}^0) = \Omega(\mathbf{L}^0; S)$ for an arbitrary loss matrix $\mathbf{L}^0$
For $t = 1$ **to** T **do**
  $\mathbf{L}^t = \frac{\nabla\psi(\mathbf{C}^{t-1})}{\|\nabla\psi(\mathbf{C}^{t-1})\|_\infty}$
  $(\widetilde{h}^t, \widetilde{\mathbf{C}}^t) = \Omega(\mathbf{L}^t; S)$
  $h^t = \left(1 - \frac{2}{t+1}\right)h^{t-1} + \frac{2}{t+1}\widetilde{h}^t$
  $\mathbf{C}^t = \left(1 - \frac{2}{t+1}\right)\mathbf{C}^{t-1} + \frac{2}{t+1}\widetilde{\mathbf{C}}^t$
End For
Output: $\bar{h} = h^T$

The approximation constants ρ and ρ' may in turn depend on the sample size N, the dimension d and the confidence level δ.

In Section 6, we discuss a practical plug-in based algorithm for implementing an LMO with these approximation properties. Equipped with access to such an LMO, we develop algorithms based on iterative optimization methods for minimizing ψ over $\mathcal{C}$. Our algorithms do not require direct access to the set $\mathcal{C}$, but only make use of calls to the LMO over $\mathcal{C}$.

We present four algorithms under different assumptions on the metric ψ and show convergence guarantees in each case (see Table 2 for a summary of our results). The proofs build on existing techniques for showing convergence of the respective optimization solvers, and need to additionally take into account the errors in the LMO calls.

4.1 Frank-Wolfe Algorithm for Smooth Convex Metrics

The first algorithm that we describe uses the classical Frank-Wolfe method (Frank and Wolfe, 1956) to minimize $\psi(\mathbf{C})$ over $\mathbf{C}$ for performance measures ψ that are convex and smooth over $\mathcal{C}$. Examples of performance measures with these properties include the H-mean and Q-mean in Table 1.

The key idea behind this algorithm is to sequentially linearize the objective ψ using its local gradients, and minimize the linear approximation over $\mathcal{C}$ using the LMO. The procedure, outlined in Algorithm 1, maintains iterates of confusion matrices $\mathbf{C}^t$, computes the gradient $\mathbf{L}^t = \nabla\psi(\mathbf{C}^{t-1})$ for the current iterate, invokes the LMO to solve the resulting linear minimization problem $\min_{\mathbf{C}\in\mathcal{C}_D}\langle \mathbf{L}^t, \mathbf{C}\rangle$, and updates $\mathbf{C}^t$ based on the result of the linear minimization. The minimizer $\mathbf{C}^*$ of $\psi(\mathbf{C})$ can then be approximated by a combination of the iterates $\mathbf{C}^1, \ldots, \mathbf{C}^T$, with the final classifier that achieves this confusion matrix given by a randomized combination of classifiers learned across all the iterations.

For metrics ψ that are smooth, we show that the algorithm takes $\mathcal{O}(1/\epsilon)$ calls to the LMO to reach a classifier that is $\mathcal{O}(\epsilon + c)$-optimal for a constant $c > 0$ that depends on the LMO error.

Theorem 12 (Convergence of FW algorithm)**.** *Fix $\epsilon \in (0,1)$. Let $\psi : [0,1]^d \to [0,1]$ be convex, β-smooth and L-Lipschitz w.r.t. the ℓ_2-norm. Let Ω in Algorithm 1 be a (ρ, ρ', δ)-approximate LMO for sample size m. Let $\bar{h}$ be a classifier returned by Algorithm 1 when run for T iterations. Then with probability $\geq 1 - \delta$ over draw of $S \sim D^N$, after $T = \mathcal{O}(1/\epsilon)$ iterations:*

$$\psi(\mathbf{C}[\bar{h}]) \leq \min_{\mathbf{C}\in\mathcal{C}} \psi(\mathbf{C}) + 8\beta\epsilon + 2L\rho + 4\beta\sqrt{d}\rho' \leq \min_{\mathbf{C}\in\mathcal{C}} \psi(\mathbf{C}) + \mathcal{O}(\epsilon + \rho^{\text{eff}}),$$

where $\rho^{\text{eff}} = \rho + \sqrt{d}\rho'$.

Proof. See Appendix A.4. □

The proof derives a version of the convergence guarantee for the Frank-Wolfe method (Jaggi, 2013) which is robust to errors in the gradients and confusion matrix estimates.

In Figure 5a, we illustrate the trajectory taken by the Frank-Wolfe algorithm in minimizing the H-mean loss ψ^{HM} in Table 1. Notice that the linear minimization outputs $\widetilde{\mathbf{C}}^t$ lie on the boundary of $\mathcal{C}$, while the averaged confusion matrix iterates $\mathbf{C}^t$ lie in the interior. Also note that because `NormImbal` distribution we use for this illustration has significant class imbalance, the minimizer for the 0-1 loss incurs a large H-mean loss. In contrast, Algorithm 1 converges to a confusion matrix with substantially better H-mean loss.

4.2 Gradient Descent-Ascent Algorithm for Non-smooth Convex Metrics

The next algorithm we propose is designed for performance measures ψ that are convex, but *not necessarily smooth*, such as the min-max metric in Table 1. We make use of the "three player" framework proposed by Narasimhan et al. (2019) and provide a slight variant of the "oracle-based algorithm" in their paper.

As a first step, we decouple the confusion matrix $\mathbf{C}$ from the function ψ in OP1* by introducing auxiliary slack variables $\boldsymbol{\xi} \in \Delta_d$, and arrive at the following equivalent problem:

$$\min_{\mathbf{C} \in \mathcal{C}} \psi(\mathbf{C}) = \min_{\mathbf{C} \in \mathcal{C}, \boldsymbol{\xi} \in \Delta_d, \boldsymbol{\xi} = \mathbf{C}} \psi(\boldsymbol{\xi}), \tag{1}$$

where we constraint the slack variables ξ to be equal to the confusion matrix $\mathbf{C}$. We define the Lagrangian for the above problem introducing multipliers $\boldsymbol{\lambda} \in \mathbb{R}^d$ for the d equality constraints:

$$\mathcal{L}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}) = \psi(\boldsymbol{\xi}) + \langle \boldsymbol{\lambda}, \mathbf{C} - \boldsymbol{\xi} \rangle, \tag{2}$$

and re-formulate (1) as an equivalent min-max problem where we minimize the Lagrangian over $\boldsymbol{\xi}$ and $\mathbf{C}$, and maximize it over the Lagrange multipliers $\boldsymbol{\lambda}$:

$$\min_{\mathbf{C} \in \mathcal{C}} \psi(\mathbf{C}) = \min_{\mathbf{C} \in \mathcal{C}, \boldsymbol{\xi} \in [0,1]^d} \max_{\boldsymbol{\lambda} \in \mathbb{R}^d} \mathcal{L}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}). \tag{3}$$

The minimizer of $\psi(\mathbf{C})$ over $\mathbf{C}$ can be then obtained by finding a saddle point of the above min-max problem. To this end, we first notice that the Lagrangian $\mathcal{L}$ is linear in $\mathbf{C}$, convex in $\boldsymbol{\xi}$ and linear in $\boldsymbol{\lambda}$. Following Narasimhan et al. (2019), we maintain iterates $\mathbf{C}^t, \boldsymbol{\xi}^t$ and $\boldsymbol{\lambda}^t$ and at each iteration, perform a full minimization of $\mathcal{L}$ using a call to the LMO, perform gradient descent updates on $\boldsymbol{\xi}$, and perform gradient ascent updates on $\boldsymbol{\lambda}$. We constrain $\boldsymbol{\xi}$ to be within the probability simplex Δ_d, and for technical reasons, also constrain $\boldsymbol{\lambda}$ to be within a bounded set Λ, both of which are accomplished using projection operations.

The resulting gradient descent-ascent procedure, outlined in Algorithm 2 can be shown to converge to an approximate saddle point of (3). In fact, one can further show that with $\mathcal{O}(\log(d)/\epsilon^2)$ calls to the LMO, the algorithm finds a classifier that is $\mathcal{O}(\epsilon + c)$-optimal for ψ, for some constant $c > 0$ that depends on the LMO errors:[†]

[†]Narasimhan et al. (2019) point out that the min-max formulation in (3) can be used to re-derive the Frank-Wolfe based procedure in Algorithm 1. Specifically, by defining $\omega(\mathbf{C}, \lambda) = \min_{\boldsymbol{\xi} \in [0,1]^d} \mathcal{L}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda})$, and reformulate (OP2*) as the equivalent min-max problem $\min_{\mathbf{C} \in \mathcal{C}} \max_{\boldsymbol{\lambda} \in \mathbb{R}^d} \omega(\mathbf{C}, \lambda)$, the Frank-Wolfe based algorithm can be shown to minimize ω using a LMO over $\mathbf{C} \in \mathcal{C}$ and maximize it over $\boldsymbol{\lambda} \in \mathbb{R}^d$ by applying a Follow-The-Leader (FTL) update (Abernethy and Wang, 2017).

Algorithm 2 Gradient Descent-Ascent (GDA) Algorithm for OP1 with Non-smooth Convex ψ

Input: $\psi : [0,1]^d \to [0,1]$, an LMO Ω , $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, T, space of Lagrange multipliers $\Lambda \subset \mathbb{R}^d$
Parameters: Step-sizes $\omega, \omega' > 0$
Initialize: $\boldsymbol{\lambda}^0 \in \Lambda$
For $t = 0$ **to** $T-1$ **do**
  $\mathbf{L}^t = \frac{\boldsymbol{\lambda}^t}{\|\boldsymbol{\lambda}^t\|_\infty}$
  $(h^t, \mathbf{C}^t) = \Omega(\mathbf{L}^t; S)$
  $\widetilde{\boldsymbol{\xi}} = \boldsymbol{\xi}^t - \omega \nabla_{\boldsymbol{\xi}} \mathcal{L}(\mathbf{C}^t, \boldsymbol{\xi}^t, \boldsymbol{\lambda}^t); \quad \boldsymbol{\xi}^{t+1} \in \operatorname{argmin}_{\boldsymbol{\xi} \in \Delta_d} \|\boldsymbol{\xi} - \widetilde{\boldsymbol{\xi}}\|_2$
  $\widetilde{\boldsymbol{\lambda}} = \boldsymbol{\lambda}^t + \omega' \nabla_{\boldsymbol{\lambda}} \mathcal{L}(\mathbf{C}^t, \boldsymbol{\xi}^t, \boldsymbol{\lambda}^t); \quad \boldsymbol{\lambda}^{t+1} \in \operatorname{argmin}_{\boldsymbol{\lambda} \in \Lambda} \|\boldsymbol{\lambda} - \widetilde{\boldsymbol{\lambda}}\|_2$
End For
Output: $\bar{h} = \frac{1}{T} \sum_{t=1}^{T} h^t$

Theorem 13 (Convergence of GDA algorithm)**.** *Fix $\epsilon \in (0,1)$. Let $\psi : [0,1]^d \to [0,1]$ be convex and L-Lipschitz w.r.t. the ℓ_2-norm. Let Ω in Algorithm 2 be a (ρ, ρ', δ)-approximate LMO for sample size N. Let the space of Lagrange multipliers $\Lambda = \{\boldsymbol{\lambda} \in \mathbb{R}^d \mid \|\boldsymbol{\lambda}\|_2 \leq 2L\}$. Let $\bar{h}$ be a classifier returned by Algorithm 2 when run for T iterations, with step-sizes $\omega = \frac{1}{4L\sqrt{2T}}$ and $\omega' = \frac{4L}{\sqrt{2T}}$. Then with probability $\geq 1 - \delta$ over draw of $S \sim D^N$, after $T = \mathcal{O}(1/\epsilon^2)$ iterations:*

$$\psi(\mathbf{C}[\bar{h}]) \leq \min_{\mathbf{C} \in \mathcal{C}} \psi(\mathbf{C}) + \mathcal{O}\left(\epsilon + \rho^{\text{eff}}\right),$$

where $\rho^{\text{eff}} = \rho + \sqrt{d}\rho'$ and $\mathcal{O}$ hides constants independent of ϵ, ρ, ρ' and d.

Proof. See Appendix A.5. □

Figure 5b shows the trajectory of the iterates of the GDA algorithm on the same running example used to illustrate the Frank-Wolfe based algorithm. Notice that the GDA algorithm converges to an optimal confusion matrix (classifier) for the problem.

4.3 Ellipsoid Algorithm for Non-smooth Convex Metrics

Building on the Lagrangian dual formulation described above, we next design an approach based on the classical ellipsoid algorithm (Boyd and Vandenberghe, 2004), which for convex (non-smooth) performance measures ψ, requires only $\mathcal{O}(d^2 \log(d/\epsilon))$ calls to the LMO to reach an $\mathcal{O}(\epsilon + c)$-optimal classifier. Note that unlike the two previous algorithms, the number of LMO calls in this case has a logarithmic dependence on $1/\epsilon$, but at the cost of a stronger dependence on dimension d. So for problems where d is small, we expect this approach to enjoy faster convergence.

We begin by defining the Lagrange dual function for given multipliers $\boldsymbol{\lambda}$:

$$f(\boldsymbol{\lambda}) = \min_{\mathbf{C} \in \mathcal{C}, \boldsymbol{\xi} \in \Delta_d} \mathcal{L}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}).$$

Because f is concave in $\boldsymbol{\lambda}$, we can employ the ellipsoid algorithm to efficiently maximize f over $\boldsymbol{\lambda}$ and thus solve OP1. Each step of the algorithm requires computing a super-gradient for f at the current iterate $\boldsymbol{\lambda}^t$, which serves as a hyper-plane separating $\boldsymbol{\lambda}^t$ from the maximizer of f. For

Algorithm 3 Ellipsoid Algorithm for OP1 with Non-smooth Convex ψ

1: **Input:** $\psi : [0,1]^d \to [0,1]$, an LMO Ω, $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}, T$
2: **Parameters:** Initial ellipsoid radius a
3: **Initialize:** $\widetilde{\boldsymbol{\lambda}}^0 = \mathbf{0}_d$, $\widetilde{\mathbf{A}}^0 = a^2\mathbf{I}_d$
4: **For** $t = 0$ **to** $T-1$ **do**
5: **If** $\|\widetilde{\boldsymbol{\lambda}}^t\|_2 > a$:
6: $\mathbf{A}^{t+1}, \boldsymbol{\lambda}^{t+1} = \text{JLE}(\mathbf{A}^t, \boldsymbol{\lambda}^t, -\boldsymbol{\lambda}^t)$
7: $h^t, \mathbf{C}^t = h^0, \mathbf{C}^0$
8: **Else:**
9: $\mathbf{A}^t, \boldsymbol{\lambda}^t = \widetilde{\mathbf{A}}^t, \widetilde{\boldsymbol{\lambda}}^t$
10: $(h^t, \mathbf{C}^t) = \Omega(\boldsymbol{\lambda}^t, S)$
11: $\boldsymbol{\xi}^t = \operatorname{argmin}_{\boldsymbol{\xi} \in \Delta_d} \psi(\boldsymbol{\xi}) - \langle \boldsymbol{\lambda}^t, \boldsymbol{\xi} \rangle$
12: $\mathbf{A}^{t+1}, \boldsymbol{\lambda}^{t+1} = \text{JLE}(\mathbf{A}^t, \boldsymbol{\lambda}^t, \mathbf{C}^t - \boldsymbol{\xi}^t)$
13: **End For**
14: $\boldsymbol{\alpha}^* \in \operatorname{argmin}_{\boldsymbol{\alpha} \in \Delta_T} \psi\left(\sum_{t=0}^{T-1} \alpha_t \mathbf{C}^t\right)$
15: **Ouput:** $\bar{h} = \sum_{t=0}^{T-1} \alpha_t^* h^t$

Algorithm 3(a) John-Lowner Ellipsoid (JLE) Construction

1: **Input:** Positive-definite matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$, $\boldsymbol{\lambda}$, $\mathbf{w}$
2: **Output:** $\mathbf{A}', \boldsymbol{\lambda}'$ that parameterizes the smallest ellipsoid such that:

$$E(\boldsymbol{\lambda}', \mathbf{A}') \supseteq E(\boldsymbol{\lambda}, \mathbf{A}) \cap \{\mathbf{x} : (\mathbf{x} - \boldsymbol{\lambda})^\top \mathbf{w} \geq 0\}$$

where $E(\boldsymbol{\lambda}, \mathbf{A}) = \{\mathbf{x} : (\mathbf{x} - \boldsymbol{\lambda})^\top (\mathbf{A})^{-1} (\mathbf{x} - \boldsymbol{\lambda}) \leq 1\}$
3: $t = \frac{1}{d+1}, a = \frac{1}{(1-t)^2}, b = \frac{1-2t}{(1-t)^2}$
4: $\widetilde{\mathbf{w}} = \frac{A^{1/2}\mathbf{w}}{\|A^{1/2}\mathbf{w}\|_2}$
5: $\mathbf{B}^{-1} = a\widetilde{\mathbf{w}}\widetilde{\mathbf{w}}^\top + b(I - \widetilde{\mathbf{w}}\widetilde{\mathbf{w}}^\top)$
6: $\boldsymbol{\lambda}' = \boldsymbol{\lambda} + t\mathbf{A}^{\frac{1}{2}}\widetilde{\mathbf{w}}$
7: $(\mathbf{A}')^{-1} = \mathbf{A}^{-1/2}\mathbf{B}^{-1}\mathbf{A}^{-1/2}$
8: **Return** $\mathbf{A}', \boldsymbol{\lambda}'$

this, we find the minimizers $\mathbf{C}^t \in \operatorname{argmin}_{\mathbf{C} \in \mathcal{C}} \langle \boldsymbol{\lambda}^t, \mathbf{C} \rangle$ and $\boldsymbol{\xi}^t \in \operatorname{argmin}_{\boldsymbol{\xi} \in \Delta_d} \psi(\boldsymbol{\xi}) - \langle \boldsymbol{\lambda}^t, \boldsymbol{\xi} \rangle$; an application of Danskin's theorem (Danskin, 2012) then gives us that $\mathbf{C}^t - \boldsymbol{\xi}^t = \nabla_{\boldsymbol{\lambda}} \mathcal{L}(\mathbf{C}^t, \boldsymbol{\xi}^t, \boldsymbol{\lambda})$ is a super-gradient for f at $\boldsymbol{\lambda}^t$. Note that the minimization over $\mathbf{C}$ can be performed (approximately) by calling the LMO Ω, and the minimization over $\boldsymbol{\xi}$ is a simple convex program.

The algorithm uses the (approximate) super-gradient obtained above to maintain an ellipsoid containing a solution that approximately maximises $f(\cdot)$ (with the current iterate $\boldsymbol{\lambda}^t$ serving as the center of the ellipsoid), and iteratively shrinks its volume until we reach a small-enough region enclosing the maximizer. In Algorithm 3, we outline the details of the procedure. Lines 5-7 of the algorithm are added to ensure that the iterates $\boldsymbol{\lambda}^t$ never leave the initial ball.

The main loop of Algorithm 3 gives us a solution $\boldsymbol{\lambda}$ that is close to the optimal dual solution. All that remains is to convert this to a solution for the primal problem in OP1*. For this, we adopt an approach from Lee et al. (2015), which uses the fact that the algorithm maintains a subset of solutions obtained from convex combinations of the confusion matrix iterates $\operatorname{conv}\left(\mathbf{C}^0, \ldots, \mathbf{C}^{T-1}\right)$, each of which is a primal-optimal solution. Furthermore because the ellipsoid algorithm returns a solution from this set which is (approximately) dual-optimal, we have that:

$$\max_{\boldsymbol{\lambda} \in \mathbb{R}^d} \min_{\mathbf{C} \in \mathcal{C}, \boldsymbol{\xi} \in \Delta_d} \mathcal{L}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}) \sim \max_{\boldsymbol{\lambda} \in \mathbb{R}^d} \min_{\substack{\mathbf{C} \in \operatorname{conv}\left(\mathbf{C}^0, \ldots, \mathbf{C}^{T-1}\right) \\ \boldsymbol{\xi} \in \Delta_d}} \mathcal{L}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}).$$

An application of min-max theorem then gives us that an approximate primal-optimal solution can be found by solving:

$$\min_{\substack{\mathbf{C} \in \operatorname{conv}\left(\mathbf{C}^0, \ldots, \mathbf{C}^{T-1}\right) \\ \boldsymbol{\xi} \in \Delta_d}} \max_{\boldsymbol{\lambda} \in \mathbb{R}^d} \mathcal{L}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}) \;=\; \min_{\mathbf{C} \in \operatorname{conv}\left(\mathbf{C}^0, \ldots, \mathbf{C}^{T-1}\right)} \psi(\mathbf{C}),$$

which amounts to solving a convex program with no further calls to the LMO and does not require further access to the training data. Line 14 of Algorithm 3 describes this post-processing step.

Theorem 14 (Convergence of Ellipsoid algorithm)**.** *Fix $\epsilon \in (0, 1)$. Let $\psi : [0, 1]^d \to [0, 1]$ be convex and L-Lipschitz w.r.t. the ℓ_2 norm. Let Ω in Algorithm 3 be a (ρ, ρ', δ)-approximate LMO for sample size N. Let $\bar{h}$ be the classifier returned by Algorithm 3 when run for T iterations with initial radius $a = 2L$. Then with probability $\geq 1 - \delta$ over draw of $S \sim D^N$, after $T = \mathcal{O}\left(d^2 \log\left(d/\epsilon\right)\right)$ iterations:*

$$\psi(\mathbf{C}[\bar{h}]) \;\leq\; \min_{\mathbf{C} \in \mathcal{C}} \psi(\mathbf{C}) \;+ \mathcal{O}\left(\epsilon + \rho^{\text{eff}}\right),$$

where $\rho^{\text{eff}} = \rho + \sqrt{d}\rho'$ and the $\mathcal{O}$ notation hides constant factors independent of ρ, ρ', ϵ, d.

Proof. See Appendix A.6. □

Figure 5c illustrates the trajectory taken by the LMO iterates and the final confusion matrix for the running example, and demonstrates the convergence of the algorithm to an optimal classifier.

4.4 Bisection Algorithm for Ratio-of-linear Metrics

The final algorithm we describe in this section uses the bisection method (Boyd and Vandenberghe, 2004) and is designed for ratio-of-linear performance metrics that can be written in the form $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$ for some $\mathbf{A}, \mathbf{B} \in \mathbb{R}^d$, such as the micro F_1-measure in Example 4.

For these performance measures, it is easy to see that:

$$\min_{\mathbf{C} \in \mathcal{C}} \psi(\mathbf{C}) \geq \gamma \iff \min_{\mathbf{C} \in \mathcal{C}} \langle \mathbf{A} - \gamma \mathbf{B}, \mathbf{C} \rangle \geq 0.$$

Thus, to test whether the optimal value of ψ is greater than γ, one can simply solve the linear minimization problem $\min_{\mathbf{C} \in \mathcal{C}} \langle \mathbf{A} - \gamma \mathbf{B}, \mathbf{C} \rangle$ and test the value of ψ at the resulting minimizer. Based on this observation, one can employ the bisection method to conduct a binary search for the minimal value (and the minimizer) of $\psi(\mathbf{C})$ using only a linear minimization subroutine.

Algorithm 4 Bisection Algorithm for OP1 with Ratio-of-linear ψ

Input: $\psi : [0,1]^d \to [0,1]$ s.t. $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$ with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^d$
  an LMO Ω, $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, T
Initialize: $\alpha^0 = 0, \beta^0 = 1$, arbitrary classifier h^0
For $t = 1$ to T **do**
  $\gamma^t = (\alpha^{t-1} + \beta^{t-1})/2$
  $\mathbf{L}^t = \frac{\mathbf{A} - \gamma^t \mathbf{B}}{\|\mathbf{A} - \gamma^t \mathbf{B}\|_2}$
  $(g^t, \mathbf{C}^t) = \Omega(\mathbf{L}^t; S)$
  If $\psi(\mathbf{C}^t) \leq \gamma^t$ **then** $\alpha^t = \alpha^{t-1}, \ \beta^t = \gamma^t, \ h^t = g^t$
    else $\alpha^t = \gamma^t, \ \beta^t = \beta^{t-1}, \ h^t = h^{t-1}$
End For
Output: $\bar{h} = h^T$

As outlined in Algorithm 4, our proposed approach maintains a confusion matrix $\mathbf{C}^t$ implicitly via classifier h^t, together with lower and upper bounds α^t and β^t on the minimal value of ψ. At each iteration, it determines whether this minimal value is greater than the midpoint γ^t of these bounds using a call to the LMO, and then update $\mathbf{C}^t$ and α^t, β^t accordingly. Since for ratio-of-linear performance measures there is always a deterministic classifier achieving the optimal performance (see Proposition 8), here it suffices to maintain deterministic classifiers h^t.

Like the previous ellipsoid-based algorithm, the bisection algorithm also enjoys a logarithmic convergence rate:[‡]

Theorem 15 (Convergence of Bisection algorithm)**.** *Fix $\epsilon \in (0,1)$. Let $\psi : [0,1]^d \to [0,1]$ be such that $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$, where $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$, and $\min_{\mathbf{C} \in \mathcal{C}} \langle \mathbf{B}, \mathbf{C} \rangle = b$ for some $b > 0$. Let Ω in Algorithm 4 be a (ρ, ρ', δ)-approximate LMO for sample size N. Let $\bar{h}$ be a classifier returned by Algorithm 4 when run for T iterations. Then with probability $\geq 1 - \delta$ over draw of $S \sim D^N$, after $T = \log(1/\epsilon)$ iterations:*

$$\psi(\mathbf{C}[\bar{h}]) \ \leq \ \min_{\mathbf{C} \in \mathcal{C}} \psi(\mathbf{C}) \ + \ \mathcal{O}\left(\epsilon + \rho^{\text{eff}}\right),$$

where $\rho^{\text{eff}} = \rho + \sqrt{d}\rho'$ and the $\mathcal{O}$ notation hides constant factors independent of ρ, ρ', ϵ and d.

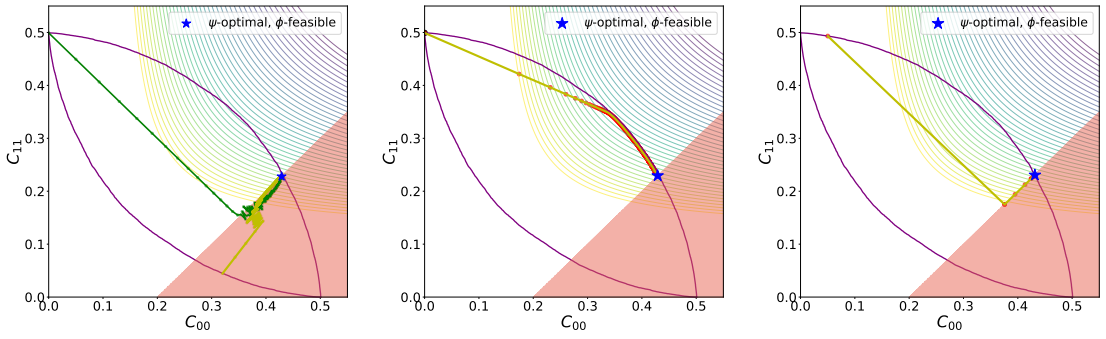
Proof. See Appendix A.7. □

5. Algorithms for Constrained Problems

We next present iterative algorithms for solving the constrained learning problem in OP2, which as noted earlier, can be viewed as a minimization problem over $\mathcal{C}$:

$$\min_{\mathbf{C} \in \mathcal{C}, \boldsymbol{\phi}(\mathbf{C}) \leq \mathbf{0}} \psi(\mathbf{C}). \tag{OP2*}$$

[‡]In fact, the bisection algorithm can be viewed as a special case of the ellipsoid algorithm in one dimension (Boyd and Vandenberghe, 2004).



(a) SplitFW (b) ConGDA (c) ConEllipsoid

Figure 6: Illustration of the Split Frank-Wolfe (Algorithm 1), Constrained GDA (Algorithm 6) and Constrained Ellipsoid (Algorithm 7) algorithms in minimizing the H-mean loss ψ^{HM} subject to the constraint $C_{00} - C_{11} \geq 0.2$ on the `NormBal` distribution in Figure 2b. The figures contain the space of achievable confusion matrices $\mathcal{C}$ (with purple colored boundary), along with the contours of ψ^{HM}. The trajectory of the averaged confusion matrix $\mathbf{C}[\bar{h}^t]$ for the averaged classifier is shown in green, where $\bar{h}^t = h^t$ for Frank-Wolfe, $\bar{h}^t = \frac{1}{t}\sum_{\tau=1}^{t} h^t$ for GDA, and $\bar{h}^t = \frac{1}{t}\sum_{\tau=1}^{t} \alpha_\tau^* h^\tau$ for ellipsoid, with the optimal coefficients $\boldsymbol{\alpha}^* \in \operatorname{argmin}_{\boldsymbol{\alpha}\in\Delta_t:\, \boldsymbol{\phi}(\sum_{\tau=1}^{t}\alpha_\tau \mathbf{C}^\tau)\leq \mathbf{0}} \psi\left(\sum_{\tau=1}^{t}\alpha_\tau \mathbf{C}^\tau\right)$ computed for iterates $1, \ldots t$. For SplitFW, we additionally plot the set of feasible confusion matrices $\mathcal{F}$ that satisfy the constraint (shaded red region), along with the trajectory of the the averaged auxiliary variables $\mathbf{F}^t$ (gold). The algorithms can be seen to converge to an optimal feasible solution.

As in the previous section, we will assume access to an LMO with the properties in Definition 11.

A simple approach to solving OP2 for convex ψ's and ϕ's is to formulate an equivalent convex-concave saddle point problem in terms of its Lagrangian:

$$\min_{\mathbf{C}\in\mathcal{C}} \max_{\boldsymbol{\lambda}\in\mathbb{R}^K_+} \psi(\mathbf{C}) + \sum_{k=1}^{K} \lambda_k \phi_k(\mathbf{C}) \;=\; \max_{\boldsymbol{\lambda}\in\mathbb{R}^K_+} \underbrace{\min_{\mathbf{C}\in\mathcal{C}} \psi(\mathbf{C}) + \sum_{k=1}^{K} \lambda_k \phi_k(\mathbf{C})}_{\nu(\lambda)},$$

where λ_k is the Lagrange multiplier for constraint ϕ_k, and we use strong duality to exchange the 'min' and 'max'. For a fixed $\boldsymbol{\lambda}$, the minimization over $\mathbf{C}$ is an unconstrained convex problem in $\mathbf{C}$. This resembles OP1 and can be solved with any of Algorithm 1–3 proposed in the previous section. One can therefore apply a standard gradient ascent procedure to maximize the dual function $\nu(\boldsymbol{\lambda})$, where the gradients w.r.t. $\boldsymbol{\lambda}$ can be computed by solving the minimization of $\mathbf{C}$. However, this vanilla dual-ascent approach does not enjoy strong convergence guarantees because of the multiple levels of nesting. For example, with the Frank-Wolfe based algorithm (Algorithm 1) for the inner minimization, this procedure would take $\mathcal{O}(1/\epsilon^3)$ calls to the LMO to reach an $\mathcal{O}(\epsilon)$-optimal, $\mathcal{O}(\epsilon)$-feasible solution (Narasimhan, 2018).

In what follows, we describe four algorithms for solving OP2 which require fewer calls to the LMO than the vanilla approach described above (see Table 3 for a summary of our results). The proofs build on standard techniques for showing convergence of the respective optimization solvers, but need to additionally take into account the errors in the LMO calls and need to translate the dual-optimal solution guarantees to optimality and feasibility guarantees for the primal solution.

Table 3: Algorithms for the constrained problem in OP2, with the number calls to the LMO, and the optimality gap $\psi(\mathbf{C}[\bar{h}]) - \min_{\mathbf{C}\in\mathcal{C}} \psi(\mathbf{C})$ and feasibility gap $\max_k \phi_k(\mathbf{C}[\bar{h}])$ for the returned classifier $\bar{h}$. In rows 1–3, we assume ψ is Lipschitz w.r.t. the ℓ_2-norm, and in all rows, we assume that $\phi_1, \ldots, \phi_K$ are convex and Lipschitz, and satisfy the strict feasibilty condition in Assumption 1. In row 4, $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C}\rangle}{\langle \mathbf{B}, \mathbf{C}\rangle}$ with $\min_{\mathbf{C}\in\mathcal{C}} \langle \mathbf{B}, \mathbf{C}\rangle > 0$. We denote $\bar{d} = d + K$, and $\rho^{\text{eff}} = \rho + \sqrt{\bar{d}}\rho'$.

Algorithm	Assumption on ψ	# LMO Calls	Opt. Gap	Feasibility Gap
Split Frank-Wolfe	Convex, Smooth	$\mathcal{O}\left(1/\epsilon^2\right)$	$\mathcal{O}\left(\epsilon + \sqrt{\rho^{\text{eff}}}\right)$	$\mathcal{O}\left(\epsilon + \sqrt{\rho^{\text{eff}}}\right)$
Con. GDA	Convex	$\mathcal{O}\left(K/\epsilon^2\right)$	$\mathcal{O}\left(\epsilon + \rho^{\text{eff}}\right)$	$\mathcal{O}\left(\epsilon + \rho^{\text{eff}}\right)$
Con. Ellipsoid	Convex	$\mathcal{O}\left(\bar{d}^2 \log(\bar{d}/\epsilon)\right)$	$\mathcal{O}\left(\epsilon + \rho^{\text{eff}}\right)$	$\mathcal{O}\left(\rho^{\text{eff}}\right)$
Con. Bisection	Ratio-of-linear	$\mathcal{O}\left(K \log(1/\epsilon)/\epsilon^2\right)$	$\mathcal{O}\left(\epsilon + \rho^{\text{eff}}\right)$	$\mathcal{O}\left(\epsilon + \rho^{\text{eff}}\right)$

The proposed algorithms can be seen as "constrained" counterparts to the four unconstrained algorithms described in the previous section. All our algorithms will assume that the constraints $\phi_k(\mathbf{C})$ are convex in $\mathbf{C}$. As a running example to illustrate our algorithms, we will use the task of maximizing the H-mean loss on the `NormBal` distribution described in Figure 2b, subject to the constraint that coverage on class 1 be no more than 0.3. This constraint is linear in $\mathbf{C}$ and can be written as $C_{01} + C_{11} \leq 0.3$, or equivalently re-written as $C_{00} - C_{11} \geq 0.2$ (since $\pi_1 = 0.5$).

5.1 (Split) Frank-Wolfe Algorithm for Smooth Convex Metrics

In this section, we adapt the Frank-Wolfe approach in Algorithm 1 to constrained learning problems OP2 for smooth convex metrics ψ. The key idea is to pose OP2* as an optimization problem over the intersection of two sets:

$$\min_{\mathbf{C}\in\mathcal{C}:\, \boldsymbol{\phi}(\mathbf{C})\leq \mathbf{0}} \psi(\mathbf{C}) = \min_{\mathbf{C}\in\mathcal{C}\cap\mathcal{F}} \psi(\mathbf{C}), \tag{4}$$

where $\mathcal{F} = \{\mathbf{F} \in \Delta_d \,|\, \boldsymbol{\phi}(\mathbf{F}) \leq \mathbf{0}\}$ is the set of all points in Δ_d that satisfy the K inequality constraints. While the set $\mathcal{F}$ is convex (and so is the intersection $\mathcal{C} \cap \mathcal{F}$), we will not be able to apply the classical Frank-Wolfe method to this problem as we cannot directly solve a linear minimization over the intersection $\mathcal{C} \cap \mathcal{F}$. However, we already have access to an LMO for the set $\mathcal{C}$ alone, and performing a linear minimization over the set $\mathcal{F}$ amounts to solving a straight-forward convex program. We therefore adopt the Frank-Wolfe based variant proposed by Gidel et al. (2018) for optimizing a (smooth) convex function over the intersection of two convex sets with access to linear minimization oracles for the individual sets.

To this end, we introduce auxiliary variables $\mathbf{F} \in \Delta_d$ in (4) and decouple the two constraint sets, giving us the following equivalent optimization problem:

$$\min_{\mathbf{C}\in\mathcal{C}, \mathbf{F}\in\mathcal{F}} \psi(\mathbf{C}) + \psi(\mathbf{F}) \;\text{ s.t. }\; \mathbf{C} - \mathbf{F} = 0. \tag{5}$$

We then define the augmented Lagrangian of the above problem as:

$$\mathcal{L}^{\text{aug}}(\mathbf{C}, \mathbf{F}, \boldsymbol{\lambda}) = \psi(\mathbf{C}) + \psi(\mathbf{F}) + \langle \boldsymbol{\lambda}, \mathbf{C} - \mathbf{F}\rangle + \frac{\zeta}{2}||\mathbf{C} - \mathbf{F}||_2^2, \tag{6}$$

where $\boldsymbol{\lambda}$ is a vector of Lagrange multipliers for the equality constraints and $\zeta > 0$ is a constant weight on the quadratic penalty term. We apply the approach of Gidel et al. (2018) to solve (5) by

Algorithm 5 Split Frank-Wolfe (SplitFW) Algorithm for OP2 with Smooth Convex ψ

Input: $\psi, \phi_1, \ldots, \phi_k : [0,1]^d \to [0,1]$, an LMO Ω, $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, $T \in \mathbb{N}$, $\zeta, \omega > 0$.
Initialize: $(h^0, \mathbf{C}^0) = \Omega(\mathbf{L}^0; S)$ for an arbitrary loss matrix $\mathbf{L}^0$
For $t = 1$ **to** T **do**
  $\mathbf{L}^t = \frac{\mathbf{a}^t}{\|\mathbf{a}\|_2}$, where $\mathbf{a}^t = \nabla_{\mathbf{C}} \mathcal{L}^{\text{aug}}(\mathbf{C}^{t-1}, \mathbf{F}^{t-1}, \boldsymbol{\lambda}^{t-1})$
  $(\widetilde{h}^t, \widetilde{\mathbf{C}}^t) = \Omega(\mathbf{L}^t; S)$
  $\widetilde{\mathbf{F}}^t = \operatorname{argmin}_{\mathbf{F} \in \mathcal{F}} \langle \mathbf{b}^t, \mathbf{F} \rangle$, where $\mathbf{b}^t = \nabla_{\mathbf{F}} \mathcal{L}^{\text{aug}}(\mathbf{C}^{t-1}, \mathbf{F}^{t-1}, \boldsymbol{\lambda}^{t-1})$
  $\gamma^t = \operatorname{argmin}_{\gamma \in [0,1]} \mathcal{L}^{\text{aug}}\big((1-\gamma)\mathbf{C}^{t-1} + \gamma\widetilde{\mathbf{C}}^t,\ (1-\gamma)\mathbf{F}^{t-1} + \gamma\widetilde{\mathbf{F}}^t,\ \boldsymbol{\lambda}^{t-1}\big)$
  $h^t = \big(1-\gamma^t\big)h^{t-1} + \gamma^t \widetilde{h}^t$
  $\mathbf{C}^t = (1-\gamma^t)\mathbf{C}^{t-1} + \gamma^t \widetilde{\mathbf{C}}^t$
  $\mathbf{F}^t = (1-\gamma^t)\mathbf{F}^{t-1} + \gamma^t \widetilde{\mathbf{F}}^t$
  $\boldsymbol{\lambda}^t = \boldsymbol{\lambda}^{t-1} + \frac{\omega}{t}(\mathbf{C}^t - \mathbf{F}^t)$
End For
Output: $\bar{h} = h^{t_*}$ and $\bar{\mathbf{C}} = \mathbf{C}^{t_*}$, where $t_* = \operatorname{argmin}_{t > T/2} ||\mathbf{C}^t - \mathbf{F}^t||_2^2$

using a gradient ascent step to maximize $\mathcal{L}^{\text{aug}}$ over $\boldsymbol{\lambda}$, a linear minimization step for $\mathbf{C}$ over $\mathcal{C}$, and a linear minimization step for $\mathbf{F}$ over $\mathcal{F}$.

This procedure, outlined in Algorithm 5, is guaranteed to converge to an optimal feasible classifier under the assumption that there exists a confusion matrix which is strictly feasible.

Assumption 1 (Strict feasibility)**.** *For some $r > 0$, there exists a confusion matrix $\mathbf{C}' \in \mathcal{C}$ such that $\max_{k \in [K]} \phi_k(\mathbf{C}') \leq -r$.*

Theorem 16 (Convergence of SplitFW algorithm)**.** *Fix $\epsilon > 0$. Let $\psi : [0,1]^d \to [0,1]$ be convex, β-smooth and L-Lipschitz w.r.t. the ℓ_2-norm, and let $\phi_1, \ldots, \phi_K : [0,1]^d \to [-1,1]$ be convex and L-Lipschitz w.r.t. the ℓ_2-norm. Let Ω in Algorithm 5 be a (ρ, ρ', δ)-approximate LMO for sample size N. Let $\bar{h}$ be a classifier returned by Algorithm 5 when run for T iterations with some $\zeta > 0$. Let the strict feasibility condition in Assumption 1 hold for radius $r > 0$. Then, with probability $\geq 1 - \delta$ over draw of $S \sim D^N$, after $T = \mathcal{O}(1/\epsilon^2)$ iterations:*

$$\textbf{\textit{Optimality:}}\ \ \psi(\mathbf{C}[\bar{h}]) \ \leq \min_{\mathbf{C} \in \mathcal{C}, \phi_k(\mathbf{C}) \leq 0, \forall k} \psi(\mathbf{C}) \ + \ \mathcal{O}\left(\epsilon + \sqrt{\rho^{\text{eff}}}\right);$$

$$\textbf{\textit{Feasibility:}}\ \ \phi_k(\mathbf{C}[\bar{h}]) \ \leq \ \mathcal{O}\left(\epsilon + \sqrt{\rho^{\text{eff}}}\right),\ \forall k \in [K].$$

where $\rho^{\text{eff}} = \rho + \sqrt{d}\rho'$ and the $\mathcal{O}$ notation hides constant factors independent of ρ, ρ', T, d and K for small enough ρ, ρ' and large T.

Proof. See Appendix A.8 □

Unlike the Frank-Wolfe based algorithm for the unconstrained problem (see Theorem 12) which needed only $\mathcal{O}(1/\epsilon)$ calls to the LMO to reach an $\mathcal{O}(\epsilon + c)$-optimal solution, the proposed algorithm for handling constraints requires $\mathcal{O}(1/\epsilon^2)$ calls to reach an $\mathcal{O}(\epsilon + c)$-optimal, feasible solution.

Figure 6a illustrates the trajectories of the algorithm applied to the previously described running example. As seen, both the iterates $\mathbf{C}^t$ and $\mathbf{F}^t$, representing the achievable and feasible confusion matrices respectively, are seen to converge to a solution that is optimal and feasible for the problem.

5.2 Gradient Descent-Ascent Algorithm for Non-smooth Convex Metrics

Next, we modify the gradient descent-ascent approach in Algorithm 2 to handle constraints. Our proposal is a slight variant of the oracle-based algorithm in Narasimhan et al. (2019) for optimizing with constraints. As before, we introduce slack variables $\boldsymbol{\xi} \in \Delta_d$ to decouple the functions $\psi, \phi_1, \ldots, \phi_K$ from the confusion matrix $\mathbf{C}$, and re-write OP2* as:

$$\min_{\mathbf{C} \in \mathcal{C}:\, \boldsymbol{\phi}(\mathbf{C}) \leq \mathbf{0}} \psi(\mathbf{C}) = \min_{\substack{\mathbf{C} \in \mathcal{C},\, \boldsymbol{\xi} \in \Delta_d \\ \boldsymbol{\xi} = \mathbf{C},\, \phi_k(\boldsymbol{\xi}) \leq 0, \forall k}} \psi(\boldsymbol{\xi}) \tag{7}$$

We then define the Lagrangian for the above problem with multipliers $\boldsymbol{\lambda} \in \mathbb{R}^d$ for the d equality constraints and $\boldsymbol{\mu} \in \mathbb{R}^K_+$ for the K inequality constraints:

$$\mathcal{L}^{\text{con}}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \psi(\boldsymbol{\xi}) + \langle \boldsymbol{\lambda}, \mathbf{C} - \boldsymbol{\xi} \rangle + \langle \boldsymbol{\mu}, \boldsymbol{\phi}(\boldsymbol{\xi}) \rangle, \tag{8}$$

and re-formulate (7) as the following min-max problem:

$$\min_{\mathbf{C} \in \mathcal{C}, \phi_k(\mathbf{C}) \leq 0, \forall k} = \min_{\mathbf{C} \in \mathcal{C},\, \boldsymbol{\xi} \in \Delta_d} \max_{\boldsymbol{\lambda} \in \mathbb{R}^d, \boldsymbol{\mu} \in \mathbb{R}^K_+} \mathcal{L}^{\text{con}}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}). \tag{9}$$

The gradient descent-ascent procedure for solving an approximate saddle point of (9) is shown in Algorithm 2 and enjoys the following convergence guarantee for a convex, non-smooth metric ψ:

Theorem 17 (Convergence of ConGDA algorithm)**.** *Fix $\epsilon \in (0, 1)$. Let $\psi : [0, 1]^d \to [-1, 1]$ and $\phi_1, \ldots, \phi_K : [0, 1]^d \to [-1, 1]$ be convex and L-Lipschitz w.r.t. the ℓ_2-norm. Let Ω in Algorithm 6 be a (ρ, ρ', δ)-approximate LMO for sample size N. Suppose the strict feasibility condition in Assumption 1 holds for radius $r > 0$. Let the space of Lagrange multipliers $\Lambda = \{\boldsymbol{\lambda} \in \mathbb{R}^d \,|\, \|\boldsymbol{\lambda}\|_2 \leq 2L(1 + 1/r)\}$, and $\Xi = \{\boldsymbol{\mu} \in \mathbb{R}^K_+ \,|\, \|\boldsymbol{\mu}\|_1 \leq 2/r\}$. Let $\bar{h}$ be a classifier returned by Algorithm 6 when run for T iterations, with step-sizes $\omega = \frac{1}{\bar{L}\sqrt{2T}}$ and $\omega' = \frac{\bar{L}}{(1+2\sqrt{K})\sqrt{2T}}$, where $\bar{L} = 4(1 + 1/r)L + 2/r$. Then with probability $\geq 1 - \delta$ over draw of $S \sim D^N$, after $T = \mathcal{O}(K/\epsilon^2)$ iterations:*

$$\textbf{\textit{Optimality:}}\ \ \psi(\mathbf{C}[\bar{h}]) \ \leq \ \min_{\mathbf{C} \in \mathcal{C}:\, \boldsymbol{\phi}(\mathbf{C}) \leq \mathbf{0}} \psi(\mathbf{C}) \ + \ \mathcal{O}\left(\epsilon + \rho^{\text{eff}}\right);$$

$$\textbf{\textit{Feasibility:}}\ \ \phi_k(\mathbf{C}[\bar{h}]) \ \leq \ \mathcal{O}\left(\epsilon + \rho^{\text{eff}}\right), \forall k \in [K].$$

where $\rho^{\text{eff}} = \rho + \sqrt{d}\rho'$ and the $\mathcal{O}$ notation hides constant factors independent of ρ, ρ', T, d and K.

Proof. See Appendix A.9. □

Figure 6b shows the trajectory of the iterates of the algorithm on the same running example used for the SplitFW algorithm. The algorithm is seen to converge to an optimal-feasible classifier.

Algorithm 6 Constrained GDA (ConGDA) Algorithm for OP2 with Non-smooth Convex ψ

1: **Input:** $\psi, \phi_1, \ldots, \phi_K : [0,1]^d \to [0,1]$, an LMO Ω, $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, T, space of Lagrange multipliers $\Lambda \subset \mathbb{R}^d, \Xi \subset \mathbb{R}^K_+$
2: **Parameters:** Step-sizes $\omega_{\boldsymbol{\xi}}, \omega_{\boldsymbol{\lambda}}, \omega_{\boldsymbol{\mu}} > 0$
3: **Initialize:** $(h^0, \mathbf{C}^0) = \Omega(\mathbf{L}^0; S)$ for an arbitrary loss matrix $\mathbf{L}^0$
4: **For** $t = 1$ **to** T **do**
5: $\quad \mathbf{L}^t = \frac{\boldsymbol{\lambda}^{t-1}}{\|\boldsymbol{\lambda}^{t-1}\|_2}$
6: $\quad (h^t, \mathbf{C}^t) = \Omega(\mathbf{L}^t; S)$
7: $\quad \widetilde{\boldsymbol{\xi}} = \boldsymbol{\xi}^{t-1} - \omega_{\boldsymbol{\xi}} \nabla_{\boldsymbol{\xi}} \mathcal{L}^{\text{con}}(\mathbf{C}^t, \boldsymbol{\xi}^{t-1}, \boldsymbol{\lambda}^{t-1}, \boldsymbol{\mu}^{t-1}); \quad \boldsymbol{\xi}^{t+1} \in \operatorname{argmin}_{\boldsymbol{\xi} \in [0,1]^d} \|\boldsymbol{\xi} - \widetilde{\boldsymbol{\xi}}\|_2$
8: $\quad \widetilde{\boldsymbol{\lambda}} = \boldsymbol{\lambda}^{t-1} + \omega_{\boldsymbol{\lambda}} \nabla_{\boldsymbol{\lambda}} \mathcal{L}^{\text{con}}(\mathbf{C}^t, \boldsymbol{\xi}^{t-1}, \boldsymbol{\lambda}^{t-1}, \boldsymbol{\mu}^{t-1}); \quad \boldsymbol{\lambda}^{t+1} \in \operatorname{argmin}_{\boldsymbol{\lambda} \in \Lambda} \|\boldsymbol{\lambda} - \widetilde{\boldsymbol{\lambda}}\|_2$
9: $\quad \widetilde{\boldsymbol{\mu}}^t = \boldsymbol{\mu}^{t-1} + \omega_{\boldsymbol{\mu}} \nabla_{\boldsymbol{\mu}} \mathcal{L}^{\text{con}}(\mathbf{C}^t, \boldsymbol{\xi}^{t-1}, \boldsymbol{\lambda}^{t-1}, \boldsymbol{\mu}^{t-1}); \quad \boldsymbol{\mu}^{t+1} \in \operatorname{argmin}_{\boldsymbol{\mu} \in \Xi} \|\boldsymbol{\mu} - \widetilde{\boldsymbol{\mu}}\|_2$
10: **End For**
11: **Output:** $\bar{h} = \frac{1}{T} \sum_{t=1}^{T} h^t$

5.3 Ellipsoid Algorithm for Non-smooth Convex Metrics

Our next algorithm extends the ellipsoid method in Algorithm 3 to handle constraints $\boldsymbol{\phi}(\mathbf{C}) \leq \mathbf{0}$. We use the Lagrangian $\mathcal{L}^{\text{con}}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ for the constrained problem defined in the previous section in (8), and work with its dual function f:

$$f^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \begin{cases} \min_{\mathbf{C} \in \mathcal{C}, \boldsymbol{\xi} \in \Delta_d} \mathcal{L}^{\text{con}}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) & \text{if } \boldsymbol{\mu} \geq 0 \\ -\infty & \text{otherwise} \end{cases},$$

where we note that the Lagrange multipliers $\boldsymbol{\mu}$ for the K inequality constraints are not allowed to be negative.

Following the unconstrained case, we seek to maximize the dual function over $\boldsymbol{\lambda} \in \mathbb{R}^d$ and over $\boldsymbol{\mu} \in \mathbb{R}^K_+$. Because f^{con} is concave in $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, we can employ the ellipsoid method with the JLE subroutine in Algorithm 3(a) to maximize $f^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu})$, and use a post-processing step to convert the dual solution to a near-optimal and near-feasible solution for the primal problem. As shown in Algorithm 7, at each iteration, the procedure maintains an ellipsoid containing the maximizer of f^{con}, with the current iterate $[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]$ serving as the center of the ellipsoid

Lines 5 to 10 of the algorithm simply ensure the iterate $[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]$ stays within the initial ellipsoid, and $\boldsymbol{\mu}^t$ remains non-negative. As before, to compute a super-gradient for f at a given $[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]$, we compute $\mathbf{C}^t \in \operatorname{argmin}_{\mathbf{C} \in \mathcal{C}} \langle \boldsymbol{\lambda}^t, \mathbf{C} \rangle$ and $\boldsymbol{\xi}^t \in \operatorname{argmin}_{\boldsymbol{\xi} \in \Delta_d} \psi(\boldsymbol{\xi}) - \langle \boldsymbol{\lambda}^t, \boldsymbol{\xi} \rangle + \langle \boldsymbol{\mu}^t, \boldsymbol{\phi}(\boldsymbol{\xi}) \rangle$, and evaluate $[\mathbf{C}^t - \boldsymbol{\xi}^t, \boldsymbol{\phi}(\boldsymbol{\xi}^t)]$. Note that $\mathbf{C}^t$ can be obtained via a linear minimization oracle over $\mathcal{C}$ and $\boldsymbol{\xi}^t$ is the solution of a convex program that has no dependence on the data distribution. The approximate nature of the LMO (and in turn the supergradient of f^{con}) require a modified proof from the standard ellipsoid to argue that the errors at each iteration do not add up catastrophically. The dual solution is converted to a primal-feasible solution in line 16 of the algorithm by solving a convex optimization problem that requires no access to the training data.

In Algorithm 7, the initial classifier h^0 can be any classifier, as it is the result of the LMO where the loss is the zero matrix. For the purposes of proving a convergence guarantee, we will assume that the initial classifier h^0 is strictly feasible.

Algorithm 7 Constrained Ellipsoid (ConEllipsoid) Algorithm for OP2 with Non-smooth Convex ψ

Input: $\psi : [0,1]^d \to [0,1]$, an LMO Ω, $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}, T$
Parameters: Initial ellipsoid radius a, a strictly feasible classifier h^0
Initialize: $\boldsymbol{\lambda}^0 = \mathbf{0}_d, \boldsymbol{\mu}^0 = 0$, $\mathbf{A}^0 = a^2\mathbf{I}_{d+K}$, $\mathbf{C}^0 = \mathbf{C}[h^0]$
For $t = 0$ **to** $T-1$:
  If $\|[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]\|_2 > a$:
    $\mathbf{A}^{t+1}, [\boldsymbol{\lambda}^{t+1}, \boldsymbol{\mu}^{t+1}] = \text{JLE}(\mathbf{A}^t, [\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t], [-\boldsymbol{\lambda}^t, -\boldsymbol{\mu}^t])$
    $h^t, \mathbf{C}^t = h^0, \mathbf{C}^0$; **continue**
  Else If $\boldsymbol{\mu}^t \ngeq \mathbf{0}$:
    $\mathbf{A}^{t+1}, [\boldsymbol{\lambda}^{t+1}, \boldsymbol{\mu}^{t+1}] = \text{JLE}(\mathbf{A}^t, [\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t], [\mathbf{0}_d, \text{pos}(-\boldsymbol{\mu}^t)])$, where $\text{pos}(u) = \max(u, 0)$.
    $h^t, \mathbf{C}^t = h^0, \mathbf{C}^0$; **continue**
  Else:
    $(h^t, \mathbf{C}^t) = \Omega(\boldsymbol{\lambda}^t, S)$
    $\boldsymbol{\xi}^t = \text{argmin}_{\boldsymbol{\xi} \in \Delta_d} \psi(\boldsymbol{\xi}) - \langle \boldsymbol{\lambda}^t, \boldsymbol{\xi} \rangle + \langle \boldsymbol{\mu}^t, \boldsymbol{\phi}(\boldsymbol{\xi}) \rangle$
    $\mathbf{A}^{t+1}, [\boldsymbol{\lambda}^{t+1}, \boldsymbol{\mu}^{t+1}] = \text{JLE}(\mathbf{A}^t, [\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t], [\mathbf{C}^t - \boldsymbol{\xi}^t, \boldsymbol{\phi}(\boldsymbol{\xi}^t)])$
End For
$\boldsymbol{\alpha}^* \in \underset{\boldsymbol{\alpha} \in \Delta_T : \boldsymbol{\phi}(\sum_t \alpha_t \mathbf{C}^t) \leq \mathbf{0}}{\text{argmin}} \psi\left(\sum_{t=0}^{T-1} \alpha_t \mathbf{C}^t\right)$
Ouput: $\bar{h} = \sum_{t=0}^{T-1} \alpha_t^* h^t$

Theorem 18 (Convergence of ConEllipsoid)**.** *Fix $\epsilon \in (0,1)$. Let $\psi : [0,1]^d \to [0,1], \phi_1, \ldots, \phi_K : [0,1]^d \to [-1,1]$ be convex and L-Lipschitz w.r.t. the ℓ_2 norm. Let Ω in Algorithm 6 be a (ρ, ρ', δ)-approximate LMO for sample size N. Suppose the strict feasibility condition in Assumption 1 holds for some $r > 0$. Let the initial classifier h^0 satisfy this condition, i.e. $\boldsymbol{\phi}(\mathbf{C}[h^0]) \leq -r$ and $\mathbf{C}[h^0] = \mathbf{C}^0$. Let $\bar{d} = d + K$. Let $\bar{h}$ be the classifier returned by Algorithm 7 when run for $T > 2\bar{d}^2 \log(\frac{\bar{d}}{\epsilon})$ iterations with initial radius $a > 2(L + \frac{L+1}{r})$. Then with probability $\geq 1 - \delta$ over draw of $S \sim D^N$, we have*

$$\textbf{\textit{Optimality:}}\ \ \psi(\mathbf{C}[\bar{h}]) \leq \min_{\mathbf{C} \in \mathcal{C} :\, \phi_k(\mathbf{C}) \leq 0, \forall k} \psi(\mathbf{C}) + \mathcal{O}(\epsilon + \rho^{\text{eff}});$$
$$\textbf{\textit{Feasibility:}}\ \ \phi_k(\mathbf{C}[\bar{h}]) \leq \mathcal{O}(\rho^{\text{eff}}),\ \forall k \in [K],$$

where $\rho^{\text{eff}} = \rho + \sqrt{d}\rho'$ and the $\mathcal{O}$ notation hides constant factors independent of ρ, ρ', T, d and K.

Proof. See Appendix A.10. □

The theorem above gives guarantees on the convergence of the constrained ellipsoid algorithm to the optimal feasible solution. Notice the exponential convergence rate in $1/\epsilon$ at the cost of a quadratic dependence on dimension d and number of constraints K. Figure 6c shows the trajectory of the iterates of the algorithm on the same running example used previously. The algorithm is clearly seen to converge to an optimal-feasible classifier.

5.4 Bisection Algorithm for Fractional-linear Metrics

The final constrained algorithm we describe is a straightforward extension of the bisection method in Algorithm 4 for ratio-of-linear performance measures that can be written in the form $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$

Algorithm 8 Constrained Bisection (ConBisection) Algorithm for OP2 with Ratio-of-linear ψ

Input: $\psi : [0,1]^d \to [0,1]$ s.t. $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$ with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^d$ and $\phi_1, \ldots, \phi_K : [0,1]^d \to [0,1]$
an LMO Ω, $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, T, T', ConGDA parameters: Λ, Ξ, ω and ω'
Initialize: $\alpha^0 = 0, \beta^0 = 1$, a classifier h^0 that satisfies the constraints, i.e. $\boldsymbol{\phi}(\mathbf{C}[h^0]) \leq \mathbf{0}$
For $t = 1$ to T **do**
  $\gamma^t = (\alpha^{t-1} + \beta^{t-1})/2$
  $(g^t, \mathbf{C}^t) =$ ConGDA$(\psi', \boldsymbol{\phi}, S, \Omega, T', \Lambda, \Xi, \omega, \omega')$, where $\psi'(\mathbf{C}) = \langle \mathbf{A} - \gamma^t \mathbf{B}, \mathbf{C} \rangle$
  If $\psi(\mathbf{C}^t) \geq \gamma^t$ **then** $\alpha^t = \gamma^t, \ \beta^t = \beta^{t-1}, \ h^t = h^{t-1}$
    else $\alpha^t = \alpha^{t-1}, \ \beta^t = \gamma^t, \ h^t = g^t$
End For
Output: $\bar{h} = h^T$

for some $\mathbf{A}, \mathbf{B} \in \mathbb{R}^d$. The key observation here is that testing whether the optimal solution to the constrained problem OP2* with a ratio-of-linear ψ is greater than a threshold γ is equivalent to minimizing a linear metric with constraints:

$$\min_{\mathbf{C} \in \mathcal{C}:\, \boldsymbol{\phi}(\mathbf{C}) \leq \mathbf{0}} \psi(\mathbf{C}) \geq \gamma \iff \min_{\mathbf{C} \in \mathcal{C}:\, \boldsymbol{\phi}(\mathbf{C}) \leq \mathbf{0}} \langle \mathbf{A} - \gamma \mathbf{B}, \mathbf{C} \rangle \geq 0.$$

The latter can be solved using any of constrained learning methods outlined Algorithms 5–7. Therefore one can employ the bisection method as before to conduct a binary search for the minimal value (and minimizer) of $\psi(\mathbf{C})$ by calling one of these algorithms at each step. We outline this procedure in Algorithm 8, with the ConGDA method (Algorithm 6) used for the inner minimization.
We then have the following convergence guarantee:[§]

Theorem 19 (Convergence of ConBisection algorithm)**.** *Fix $\epsilon \in (0,1)$. Let $\psi : [0,1]^d \to [0,1]$ be such that $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$, where $\mathbf{A}, \mathbf{B} \in [0,1]^d$, and $\min_{\mathbf{C} \in \mathcal{C}} \langle \mathbf{B}, \mathbf{C} \rangle = b$ for some $b > 0$. Let $\phi_1, \ldots, \phi_K : [0,1]^d \to [-1,1]$ be convex and L-Lipschitz w.r.t. the ℓ_2-norm. Let Ω in Algorithm 8 be a (ρ, ρ', δ)-approximate LMO for sample size N. Suppose the strict feasibility condition in Assumption 1 holds for some $r > 0$. Let Λ, Ξ, ω and ω' in the call to Algorithm 6 be set as in Theorem 17 with Lipschitz constant $L' = \max\{L, \|\mathbf{A}\|_2 + \|\mathbf{B}\|_2\}$. Let $\bar{h}$ be a classifier returned by Algorithm 8 when run for T outer iterations and T' inner iterations. Then with probability $\geq 1 - \delta$ over draw of $S \sim D^N$, after $T = \log(1/\epsilon)$ outer iterations and $T' = \mathcal{O}(K/\epsilon^2)$ inner iterations:*

$$\textbf{\textit{Optimality}}: \ \psi(\mathbf{C}[\bar{h}]) \leq \min_{\mathbf{C} \in \mathcal{C}:\, \boldsymbol{\phi}(\mathbf{C}) \leq \mathbf{0}} \psi(\mathbf{C}) + \mathcal{O}\left(\epsilon + \rho^{\text{eff}}\right);$$

$$\textbf{\textit{Feasibility}}: \ \phi_k(\mathbf{C}[\bar{h}]) \leq \mathcal{O}\left(\epsilon + \rho^{\text{eff}}\right), \ \forall k \in [K],$$

where $\rho^{\text{eff}} = \rho + \sqrt{d}\rho'$ and the $\mathcal{O}$ notation hides constant factors independent of ρ, ρ', T, d and K.

Proof. See Appendix A.11. □

[§]Because the inner subroutine uses the ConGDA algorithm, the rate of convergence has a dependence of $\widetilde{O}\left(1/\epsilon^2\right)$ on ϵ, which is an improvement over the $\widetilde{O}\left(1/\epsilon^3\right)$ dependence in the previous conference paper (Narasimhan, 2018).

Algorithm 9 Plug-in Based LMO

Input: Loss matrix $\mathbf{L} \in \mathbb{R}^d_+$, $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$,
Class probability model $\widehat{\boldsymbol{\eta}} : \mathcal{X} \to \Delta_n$ independent of S
Construct classifier $\widehat{g}(x) = \operatorname{argmin}^*_{j \in [n]} \sum_{i=1}^n \widehat{\eta}_i(x) L_{n(i-1)+j}$
$\widehat{\mathbf{\Gamma}} = \operatorname{vec}\big(\widehat{\mathbf{C}}^S[\widehat{g}]\big)$
Output: $\widehat{g}$, $\widehat{\mathbf{\Gamma}}$

6. Plug-in Based Linear Minimization Oracle

All the learning algorithms we have presented have assumed access to an approximate linear minimization oracle (LMO) (see Definition 11). In this section, we describe a practical plug-in based LMO with the desired approximation properties. This method seeks to approximate the Bayes optimal classifier for the given linear metric using an estimate $\widehat{\boldsymbol{\eta}} : \mathcal{X} \to \Delta_n$ of the conditional-class probability distribution $\eta_i(X) = \mathbf{P}(Y = i|X)$.

Specifically, for a flattened loss matrix $\mathbf{L} \in \mathbb{R}^d_+$, where $L_{n(i-1)+j}$ is the cost of predicting class j when the true class is i, we have from Proposition 5 that the Bayes optimal classifier is given by $h^*(x) = \operatorname{argmin}^*_{j \in [n]} \sum_{i=1}^n \eta_i(x) L_{n(i-1)+j}$. The plug-in based LMO outlined in Algorithm 9 approximates this classifier with the class probability model $\widehat{\boldsymbol{\eta}}$. The classifier and confusion matrix returned by the algorithm satisfy the LMO approximation properties laid out in Definition 11:

Theorem 20 (Regret bound for plug-in LMO)**.** *Fix $\delta \in (0, 1)$. Then with probability $\geq 1 - \delta$ over draw of sample $S \sim D^N$, for any loss matrix $\mathbf{L} \in \mathbb{R}^d$, the classifier and confusion matrix $(\widehat{g}, \widehat{\mathbf{\Gamma}})$ returned by Algorithm 9 satisfies:*

$$\langle \mathbf{L}, \mathbf{C}[\widehat{g}] \rangle \leq \min_{h : \mathcal{X} \to \Delta_n} \langle \mathbf{L}, \mathbf{C}[h] \rangle + \|\mathbf{L}\|_\infty \mathbf{E}_X \left[\left\| \widehat{\boldsymbol{\eta}}(X) - \boldsymbol{\eta}(X) \right\|_1 \right];$$

$$\|\mathbf{C}[\widehat{g}] - \widehat{\mathbf{\Gamma}}\|_\infty \leq \mathcal{O}\left(\sqrt{\frac{d \log(n) \log(N) + \log(d/\delta)}{N}} \right).$$

Proof. See Appendix A.12 □

6.1 Consistency of Proposed Algorithms with Plug-in LMO

Theorem 20 tells us that the quality of the classifier $\widehat{g}$ returned by the plug-in based LMO depends on the estimation error $\mathbf{E}_X \left[\left\| \widehat{\boldsymbol{\eta}}(X) - \boldsymbol{\eta}(X) \right\|_1 \right]$, which measures the gap between the class probability model $\widehat{\boldsymbol{\eta}}$ and the true conditional class probabilities $\boldsymbol{\eta}$. By combining this result with Theorem 12–18, we can show that the algorithms described in Sections 4 and 5, when used with the plug-in based LMO, are statistically consistent. For the sake of brevity, we present the consistency analysis for the GDA algorithm and its constrained counter-part alone. The analysis for the other algorithms follow identical steps.

Below, we present a regret bound for Algorithms 2 and 6 with Algorithm 9 as the LMO Ω. For technical reasons, we require that the class probability model $\widehat{\eta}$ is independent of the sample S, (e.g. $\widehat{\eta}$ can be learned using a sample different from S).

Corollary 21 (Regret bound for GDA algorithm)**.** *Let $\psi : [0,1]^d \to [0,1]$ be convex and L-Lipschitz w.r.t. the ℓ_2-norm. Let the LMO Ω in Algorithm 2 be a plug-in based LMO (as in Algorithm 9) with a CPE argument $\widehat{\boldsymbol{\eta}}$. Let $\bar{h}$ be a classifier returned by Algorithm 2 when run for T iterations with the parameter settings in Theorem 13. Then with probability $\geq 1-\delta$ over draw of $S \sim D^N$, after $T = \mathcal{O}(N)$ iterations:*

$$\psi(\mathbf{C}[\bar{h}]) \leq \min_{\mathbf{C} \in \mathcal{C}} \psi(\mathbf{C}) + \mathcal{O}\left(\mathbf{E}_X[\|\widehat{\boldsymbol{\eta}}(X) - \boldsymbol{\eta}(X)\|_1] + \sqrt{d}\sqrt{\frac{d\log(n)\log(N) + \log(d/\delta)}{N}}\right).$$

Corollary 22 (Regret bound for ConGDA algorithm)**.** *Let $\psi : [0,1]^d \to [0,1]$ and $\phi_1, \ldots, \phi_K : [0,1]^d \to [-1,1]$ be convex and L-Lipschitz. Let the LMO Ω in Algorithm 2 be a plug-in based LMO (as in Algorithm 9) with a CPE argument $\widehat{\boldsymbol{\eta}}$. Let $\bar{h}$ be a classifier returned by Algorithm 6 when run for T iterations with the parameter settings in Theorem 17. Then with probability $\geq 1-\delta$ over draw of $S \sim D^N$, after $T = \mathcal{O}(KN)$ iterations:*

$$\psi(\mathbf{C}[\bar{h}]) \leq \min_{\mathbf{C} \in \mathcal{C}} \psi(\mathbf{C}) + \mathcal{O}\left(\mathbf{E}_X[\|\widehat{\boldsymbol{\eta}}(X) - \boldsymbol{\eta}(X)\|_1] + \sqrt{d}\sqrt{\frac{d\log(n)\log(N) + \log(d/\delta)}{N}}\right);$$

$$\phi_k(\mathbf{C}[\bar{h}]) \leq \mathcal{O}\left(\mathbf{E}_X[\|\widehat{\boldsymbol{\eta}}(X) - \boldsymbol{\eta}(X)\|_1] + \sqrt{d}\sqrt{\frac{d\log(n)\log(N) + \log(d/\delta)}{N}}\right), \ \forall k \in [K].$$

When the class probability model $\widehat{\boldsymbol{\eta}}$ used by the LMO is learned by an algorithm that guarantees $\mathbf{E}_X[\|\widehat{\boldsymbol{\eta}}(X) - \boldsymbol{\eta}(X)\|_1] \to 0$ as $N \to \infty$, then Algorithm 2 is statistically consistent for the unconstrained problem in (OP1), and Algorithm 6 is statistically consistent for the constrained problem in (OP2). The property that the learned class probability estimation error goes to zero in the large sample limit is true for any algorithm that minimizes a strictly proper composite multiclass loss (e.g. the standard cross-entropy loss) over a suitably large function class (Vernet et al., 2011).

While our consistency results require that the samples used by the optimization method and the LMO to be drawn independently, this may be inconvenient in real-world applications where data is scarce and limited. In practice, we find that using the same sample for both the optimization method and the LMO does not hurt performance, and this is the approach we adopt in our experiments.

A practical advantage of the plug-in based LMO is that one can pre-train the class probability model $\widehat{\boldsymbol{\eta}}$ and re-use the same model each time the LMO is invoked. In practice, there are other off-the-shelf algorithms that one can use to implement the LMO, such as cost-weighted decision trees (Ting, 2002) and those based on optimizing a cost-weighted surrogate loss (e.g. Lee et al. (2004)), which require training a new classifier for each given loss vector $\mathbf{L}$. While a majority of our experiments will use a plug-in based LMO, we also explore the use of cost-weighted surrogate losses for implementing the LMO.

7. Extension to Fairness Metrics and Other Refinements

To keep the exposition concise, we have so far focused on metrics defined by a function of the overall confusion matrix $\mathbf{C}[h]$. We now discuss how the algorithms in Sections 4 and 5 can be extended to handle the group-based fairness metrics described in Section 2.2, which are defined in terms of group-specific confusion matrices (see Definition 2).

Algorithm 10 Plug-in Based LMO for Fairness Problems

Input: Loss matrix $\mathbf{L} \in \mathbb{R}^d_+$, Class prob. model $\widehat{\boldsymbol{\eta}} : \mathcal{X} \to \Delta_n$, $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$
Group assignment $A : \mathcal{X} \to [m]$
Define $\sigma(x, i, j) = mn(A(x) - 1) + n(i - 1) + j$
Construct $\widehat{g}(x) = \operatorname{argmin}_{j \in [n]} \sum_{i=1}^{n} \widehat{\eta}_i(x) L_{\sigma(x,i,j)}$
$\widehat{\mathbf{\Gamma}} = \left[\operatorname{vec}\left(\widehat{\mathbf{C}}^0[\widehat{g}]\right), \ldots, \operatorname{vec}\left(\widehat{\mathbf{C}}^{m-1}[\widehat{g}]\right)\right]$
Output: $\widehat{g}$, $\widehat{\mathbf{\Gamma}}$

7.1 Group-based Fairness Metrics

In the fairness setup we consider, each instance $x \in \mathcal{X}$ is associated with a group $A(x) \in [m]$, and the objective and constraints are defined by functions of m group-specific confusion matrices $\mathbf{C}^1[h], \ldots, \mathbf{C}^m[h]$. Note that even for binary problems where $n = 2$, the presence of multiple groups poses challenges in solving the resulting learning problems in (OP1) and (OP2). For example, a naïve approach one could take for binary labels is to construct a simple plug-in classifier for these problems that assigns a separate threshold for each group, but tuning m thresholds via a brute-force search can quickly become infeasible when m is large.

Our approach to solving the learning problems in (OP1) and (OP2) with group fairness metrics is to once again reformulate as an optimization problem over the set of achievable group-specific confusion matrices, in this case, represented by vectors of dimension $d = mn^2$.

Definition 23 (Achievable group-specific confusion matrices)**.** *Define the set of* achievable group-specific confusion matrices w.r.t. D *as:*

$$\mathcal{C}^{[m]} = \left\{ \left[\operatorname{vec}(\mathbf{C}^0[h]), \ldots, \operatorname{vec}(\mathbf{C}^{m-1}[h])\right] \middle|\ h : \mathcal{X} \to \Delta_n \right\}.$$

Algorithms 1–8 can now be directly applied to solve the resulting optimization over $\mathcal{C}^{[m]}$, at each iteration, assuming access to an oracle for approximately solving a linear minimization problem over $\mathcal{C}^{[m]}$. This linear minimization sub-problem can again be solved using a plug-in based LMO similar Algorithm 9. The details of the plug-in variant for the fairness setup are provided in Algorithm 10, where we denote the empirical group-specific confusion matrix for group a from sample $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ by:

$$\widehat{C}^a_{ij}[h] = \frac{1}{N} \sum_{\ell=1}^{N} \mathbf{1}(y_\ell = i, h(x_\ell) = j, A(x_\ell) = a)\,.$$

7.2 Succinct Confusion Matrix Representations

Before closing, we note that for simplicity, we have allowed the d-dimensional vector representation of the confusion matrix to contain all n^2 entries (or all mn^2 for fairness metrics). In practice, we only need to take into account those entries of the confusion matrix performance measures and constraints we seek to optimize depend upon. For example, the G-mean metric in Example 3 is defined on only the diagonal entries of the confusion matrix, and so the vector representation in this case needs to only contain the n diagonal entries. In fact, for some metrics, it suffices to represent

the confusion matrix using a small number of linear transformations. For example, the coverage metric described in Example 6 is defined on only the column sums of the confusion matrix, and hence the d-dimensional vector representation in this case only needs to contain the n column sums.

More generally, we can work with succinct vector representations given by linear transformations of the confusion matrices:

Definition 24 (Generalized confusion vectors)**.** *Define the set of* (achievable) generalized confusion vectors w.r.t. D *as:*

$$\mathcal{C}^{\text{gen}} = \left\{ \left[\varphi_1(\mathbf{C}^0[h], \ldots, \mathbf{C}^{m-1}[h]), \ldots, \varphi_d(\mathbf{C}^0[h], \ldots, \mathbf{C}^{m-1}[h]) \right] \middle| \; h : \mathcal{X} \to \Delta_n \right\},$$

where each $\varphi_k : [0,1]^{mn^2} \to \mathbb{R}_+$ *is a* linear map*.*

The set $\mathcal{C}^{\text{gen}}$ is convex. In the simplest case, we can have a linear map φ_k of dimension $d = mn^2$, where each coordinate picks one entry from the m confusion matrices. However, for most of the performance metrics described in Section 2.1 and 2.2, it suffices to use a a small number of $d << mn^2$ linear transformations and we can translate the corresponding learning problems in OP1 and OP2 into equivalent optimization problems over $\mathcal{C}^{\text{gen}}$. The iterative algorithms discussed in Sections 4 and 5 can then be applied to solve the resulting lower-dimensional optimization problem over $\mathbf{C}$, with the plug-in procedure in Algorithm 9 straightforwardly adapted to solve the linear minimization over $\mathcal{C}^{\text{gen}}$ at each step.

8. Experiments

We present an experimental evaluation of the algorithms presented in Sections 4 and 5 on a variety of multi-class datasets and multi-group fair classification tasks. Broadly, we cover the following:

1. We showcase on a synthetic dataset that our algorithms converge in the large sample limit to optimal (feasible) classifier (Section 8.3).
2. We demonstrate that the proposed algorithms are competitive or better than the state-of-the-art algorithms for the real-world tasks we consider (Sections 8.4–8.5).
3. We provide practical guidance on which algorithm is better suited for a given application, and investigate two different choices for the LMO (Sections 8.6–8.7).
4. We illustrate with image classification case-studies how our algorithms can be applied to tackle class-imbalance and label noise (Section 8.8).

A summary of the datasets we use is provided in Tables 4 and 5, along with the model architecture we use in each case. The details of the data pre-processing are provided in Appendix B. With the exception of the CIFAR datasets, which comes with standard train-test splits, we split all other datasets into 2/3-rd for training and 1/3-rd for testing, and repeat our experiments over multiple such random splits. All our methods were implemented in Python using PyTorch and Scikit-learn.[¶]

8.1 Baselines

In a majority of the experiments, our algorithms will use the plug-in method in Algorithm 9 for the inner linear minimization oracle, and employ logistic regression to fit a model $\widehat{\boldsymbol{\eta}} : \mathcal{X} \to \Delta_n$ to

[¶]Code available at: `https://github.com/shivtavker/constrained-classification`

Table 4: Multi-class datasets used in our experiments

Dataset	#Classes	#Train	#Test	#Features	$\frac{\min_y \pi_y}{\max_y \pi_y}$	Model
Abalone	12	2923	1254	8	0.149	Linear
PageBlock	5	3831	1642	10	0.0057	Linear
MACHO	8	4241	1818	64	0.0148	Linear
Sat-Image	6	4504	1931	36	0.408	Linear
CovType	7	406708	174304	14	0.0097	Linear
CIFAR-10-Flip	10	27500	5500	32×32	0.1	ResNet-50
CIFAR-55	55	50000	10000	32×32	0.1	ResNet-50

Table 5: Multi-group fairness datasets with binary labels used in our experiments.

Dataset	#Train	#Test	#Features	Protected Attr.	Prot. Group Frac.	Model
Communities & Crime	1395	599	132	Race (binary)	0.49	Linear
COMPAS	4320	1852	32	Gender	0.19	Linear
Law School	14558	6240	16	Race (binary)	0.06	Linear
Default	21000	9000	23	Gender	0.40	Linear
Adult	34189	14653	123	Gender	0.10	Linear

estimate the conditional-class probabilities. As baselines, we compare with methods for minimizing the standard 0-1 loss and the balanced 0-1 loss, both of which are simpler alternatives to the metrics we consider, and the state-of-the-art approach for directly optimizing with complex metrics and constraints.

(i) A plug-in classifier that predicts the class with the maximum class probability, i.e. $\operatorname{argmax}_i \widehat{\eta}_i(x)$; this method is consistent for the 0-1 loss.

(ii) A plug-in classifier that weighs the class probabilities by the inverse class priors, and predicts the class with the highest weighted probability $\operatorname{argmax}_i \frac{1}{\widehat{\pi}_i}\widehat{\eta}_i(x)$, where $\widehat{\pi}_i$ is an estimate of the prior for class i; this method is consistent for the balanced 0-1 loss.

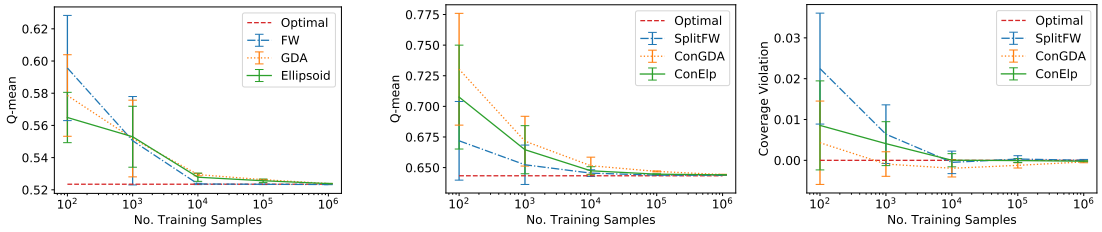
(iii) The approach of Narasimhan et al. (2019) for optimizing with complex performance metrics and constraints, available as a part of the TensorFlow Constrained Optimization (TFCO) library.[‖]

TFCO uses an optimization procedure similar to the GDA method in Algorithm 2, but instead of fitting a plug-in classifier to a pre-trained class probability model, performs online updates on surrogate approximations. Therefore one key difference between our use of plug-in classifiers and the approach taken by TFCO is that the latter is an in-training method which trains a classifier from scratch. Unlike our proposal, it does not come with consistency guarantees. It is worth noting that TFCO can be seen as a strict generalization to previous surrogate-based methods for complex evaluation metrics (Narasimhan et al., 2015a; Kar et al., 2016).

All the plug-in based methods use the same class probability estimator $\widehat{\boldsymbol{\eta}}$. We employ the same architecture as $\widehat{\boldsymbol{\eta}}$ for the model trained by TFCO.

We do not include the previous SVM$^{\text{perf}}$ method (Joachims, 2005) as a baseline because it has a running time that is exponential in the number of classes, and as shown in the previous conference version of this paper, can be prohibitively expensive to run even for a moderate number of classes (Narasimhan et al., 2015b). Moreover, this method was proposed for unconstrained problems, and does not explicitly allow for imposing constraints on metrics.

[‖] https://github.com/google-research/tensorflow_constrained_optimization



(a) Unconstrained (b) Constrained

Figure 7: Convergence of the proposed algorithms on synthetic data to (a) the Bayes optimal classifier for of the Q-mean loss, and (b) the optimal-feasible classifier for the task of minimizing Q-mean loss subject to a coverage constraint, with the Q-mean loss shown on the left and the coverage constraint violation $\max_{i\in[3]} \left|\sum_j C_{ji} - \pi_i\right| - 0.01$ shown on the right. The results are reported on the test set, and averaged over training with 5 random draws of the dataset.

8.2 Post-processing

Recall that the Frank-Wolfe, GDA and ellipsoid algorithms that we propose for convex metrics return classifiers that *randomize* over T plug-in classifiers. When implementing their constrained counterparts, we additional apply "pruning" step to the returned randomized classifier, which recomputes the convex combination of the T iterates $\mathbf{C}^1, \ldots, \mathbf{C}^T$ so that the constraints are exactly satisfied (if such a solution exists). Specifically, the final classifier is given by $\frac{1}{T}\sum_{t=1}^T \alpha_*^t\, g^t$, where $\boldsymbol{\alpha}_* \in \operatorname*{argmin}_{\boldsymbol{\alpha}\in\Delta_T:\, \sum_t \alpha^t \boldsymbol{\phi}(\mathbf{C}[g^t]) \leq \mathbf{0}} \sum_{t=1}^T \alpha^t \psi\left(\mathbf{C}[g^t]\right)$. Note that the objective here is an approximation to the true objective $\psi\left(\sum_{t=1}^T \alpha^t \mathbf{C}[g^t]\right)$, with the former upper bounding the latter when ψ is convex. This approximation to the objective allows us to compute the optimal coefficients $\boldsymbol{\alpha}_*$ by solving a simple linear program. The use of a post-processing pruning step is prescribed by the TFCO library (Cotter et al., 2019b; Narasimhan et al., 2019), and is also applied to the classifier returned by the TFCO baseline. In Appendix B.1, we provide other details such as how we choose the hyper-parameters for our algorithms and the baselines.

We additionally note that the H-mean, Q-mean and G-mean metrics we consider in our experiments can be written as functions of normalized diagonal entries of the confusion matrix: $\frac{C_{ii}}{\pi_i}, \forall i \in [n]$ (see Table 1). For these metrics, we formulate OP1 and OP2 as optimization problems over normalized confusion diagonal entries $\left[\frac{C_{11}}{\pi_1}, \ldots, \frac{C_{nn}}{\pi_n}\right]^\top \in [0,1]^n$, which is of lower-dimensional than the space of full confusion matrices. This requires a small modification to the GDA and ellipsoid algorithms, where the slack variables $\boldsymbol{\xi}$ will have to be constrained to be in $[0,1]^n$ instead of in the simplex Δ_{n^2}. Similarly, when the fairness constraints in Table 1 are enforced on binary-labeled problems, we can write the objective and constraints as functions of normalized diagonal confusion entries of group-specific confusion matrices, resulting in an optimization over vectors in $[0,1]^{2m}$.

8.3 Convergence to the Optimal Classifier

In our first set of experiments, we test the consistency behavior of the algorithms on a synthetic data set for which the Bayes optimal performance could be calculated. We use a 3-class synthetic

data set with instances in $\mathcal{X} = \mathbb{R}^2$ generated as follows: examples are chosen from class 1 with probability 0.85, from class 2 with probability 0.1 and from class 3 with probability 0.05; instances in the three classes are then drawn from multivariate Gaussian distributions with means $(1,1)^\top$, $(0,0)^\top$, and $(-1,-1)^\top$ respectively, and with the same covariance matrix $\left[\begin{smallmatrix} 5 & 1 \\ 1 & 5 \end{smallmatrix}\right]$. The conditional-class probability function $\boldsymbol{\eta} : \mathbb{R}^2 \to \Delta_3$ for this distribution is a softmax of linear functions, and can be computed in closed-form.

We first consider the unconstrained task of optimizing the Q-mean loss in Table 1, given by $\psi^{\text{QM}}(\mathbf{C}) = \left(\frac{1}{n}\sum_i \left(1 - \frac{C_{ii}}{\sum_j C_{ij}}\right)^2\right)^{1/2}$. Note that this performance metric is a smooth convex function of $\mathbf{C}$, and can be optimized with any one of the proposed Frank-Wolfe, GDA or ellipsoid methods (Algorithms 1–3). Because the metric and the distribution satisfy the conditions of Proposition 9, and the Bayes optimal classifier is of the form $h^*(x) = \operatorname{argmax}^*_{i\in[3]} w_i^* \, \eta_i(x)$, for some distribution-dependent coefficients $w_i^* \in \mathbb{R}$,. To compute the Bayes optimal classifier, we run a brute-force grid search for w_i^*.

Our algorithms use the plug-in method in Algorithm 9 for the LMO subroutine. Specifically, they fit a linear logistic regression model $\widehat{\boldsymbol{\eta}} : \mathbb{R}^2 \to \Delta_3$ to the training set, and iteratively learn a randomized combination of classifiers of the form $h(x) = \operatorname{argmax}^*_{i\in[3]} w_i \, \widehat{\eta}_i(x)$. In Figure 7a, we plot the Q-mean loss for the classifier learned by the proposed algorithms, evaluated on a test set of 10^6 examples, for different sizes of the training sample. In each case, we average the results over 5 random draws of the training sample. As seen, all three methods converge to the performance of the Bayes optimal classifier.

We next consider the task of optimizing the Q-mean loss subject to a coverage constraint, requiring the proportion of predictions made for class i to be (approximately) equal to the class prior π_i. Specifically, we constraint the max coverage deviation, $\max_{i\in[3]} \left|\sum_j C_{ji} - \pi_i\right|$ to be at most 0.01. This is a constrained problem with a convex smooth objective and a convex constraint in $\mathbf{C}$, and can be solved using the constrained counter-parts to the Frank-Wolfe, GDA and ellipsoid methods (Algorithm 5–7). Following Yang et al. (2020), we have that the optimal-feasible classifier for this problem is a randomized classifier of two classifiers $h^{1,*}(x) = \operatorname{argmax}^*_{i\in[3]} w_i^{1,*}\eta_i(x)$ and $h^{2,*}(x) = \operatorname{argmax}^*_{i\in[3]} w_i^{2,*}\eta_i(x)$, for distribution-dependent coefficients $w_i^{1,*}$ and $w_i^{2,*}$.** We compute these coefficients and the optimal randomized combination via a brute-force grid search. Figure 7b plots the Q-mean loss and the constraint violation for the three algorithms. All of them can be seen to converge to the Q-mean of the optimal-feasible classifier and to zero constraint violation.

8.4 Performance on Unconstrained Problems

We next compare the proposed algorithms for unconstrained problems on five benchmark multiclass datasets: (i) Abalone, (ii) PageBlock, (iii) CovType, (iv) SatImage and (v) MACHO. The first four were obtained from the UCI Machine Learning repository (Frank and Asuncion, 2010). The fifth dataset pertains to the task of classifying celestial objects from the Massive Compact Halo Object (MACHO) catalog using photometric time series data (Alcock et al., 2000; Kim et al., 2011). Each celestial object is described by measurements from 6059 light curves, and is categorized either as one of seven celestial objects or as a miscellaneous category.

**Proposition 10 tells us that the support of the Bayes optimal classifier randomizes over as many as $d+1$ deterministic classifiers. For the 3-class distribution we consider, $\boldsymbol{\eta}(X)$ satisfies additional continuity conditions, under which the optimal classifier can be shown to be a randomized combination of at most *two* deterministic classifier (Wang et al., 2019; Yang et al., 2020).

Table 6: Unconstrained optimization of the (convex) H-mean loss. *Lower* values are *better*. The results are averaged over 10 random train-test splits.

Dataset	Plugin [0-1]	Plugin (bal)	TFCO	FW	GDA	Ellipsoid
Abalone	1.0 ± 0.0	0.890 ± 0.038	0.824 ± 0.018	$\mathbf{0.816 \pm 0.020}$	0.818 ± 0.017	0.817 ± 0.019
Pgblk	0.416 ± 0.128	0.130 ± 0.034	0.200 ± 0.023	0.120 ± 0.028	0.130 ± 0.04	$\mathbf{0.110 \pm 0.025}$
MACHO	0.210 ± 0.043	0.130 ± 0.015	0.143 ± 0.019	0.124 ± 0.017	$\mathbf{0.124 \pm 0.015}$	0.125 ± 0.017
SatImage	0.279 ± 0.01	0.173 ± 0.008	$\mathbf{0.170 \pm 0.006}$	0.171 ± 0.007	0.173 ± 0.008	$\mathbf{0.170 \pm 0.006}$
CovType	1.0 ± 0.0	0.507 ± 0.001	0.469 ± 0.001	0.463 ± 0.001	0.463 ± 0.001	$\mathbf{0.461 \pm 0.001}$

Table 7: Unconstrained optimization of the (ratio-of-linear) micro F_1 loss. *Lower* values are *better*. The results are averaged over 10 random train-test splits.

Datasets	Plugin [0-1]	Plugin (bal)	TFCO	Bisection
Abalone	0.713 ± 0.006	0.760 ± 0.004	0.728 ± 0.012	$\mathbf{0.693 \pm 0.006}$
Pgblk	0.218 ± 0.012	0.441 ± 0.033	0.216 ± 0.018	$\mathbf{0.211 \pm 0.016}$
MACHO	$\mathbf{0.089 \pm 0.005}$	0.106 ± 0.007	0.110 ± 0.005	$\mathbf{0.089 \pm 0.005}$
SatImage	$\mathbf{0.180 \pm 0.005}$	0.185 ± 0.007	0.234 ± 0.003	$\mathbf{0.180 \pm 0.005}$
CovType	0.548 ± 0.001	0.625 ± 0.003	0.486 ± 0.001	$\mathbf{0.403 \pm 0.001}$

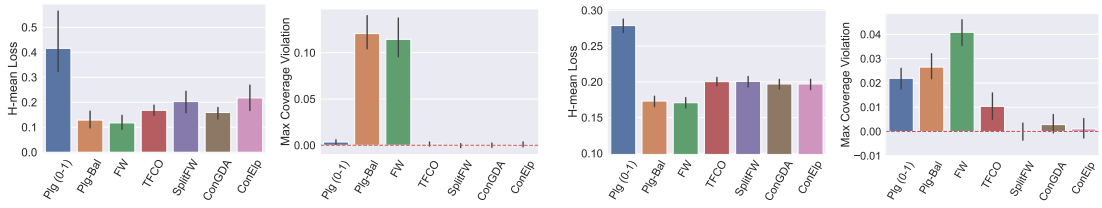
We consider two performance metrics from Table 1: (i) the H-mean metric $\psi^{\text{HM}}(\mathbf{C}) = 1 - n\left(\sum_i \frac{\sum_j C_{ij}}{C_{ii}}\right)^{-1}$ and (ii) the micro F-measure $\psi^{\text{micro}F_1}(\mathbf{C}) = 1 - \frac{2\sum_{i \neq k} C_{ii}}{2 - \sum_i C_{ki} - \sum_i C_{ik}}$, where $k \in [n]$ is a designated default class. The first metric is convex in $\mathbf{C}$, for which we compare the performances of the Frank-Wolfe, GDA, and ellipsoid algorithms (Algorithms 1–3); the second metric is ratio-of-linear in $\mathbf{C}$, and for this, we apply the bisection algorithm (Algorithm 4). Our algorithms use a plug-in based LMO with a linear logistic regression model used to estimate the conditional-class probabilities. We compare our methods with the 0-1 plug-in, balanced plug-in and TFCO baselines.

The results of optimizing the two metrics are shown in Tables 6 and 7 respectively. As expected both the 0-1 and balanced plug-in classifiers are often seen to perform poorly on the H-mean and micro F_1 metrics. For example, on the Abalone and CovType dataset, the plug-in (0-1) yields a H-mean loss of 1 as it achieves high accuracies on the higher-frequency classes at the cost of yielding zero accuracy on one or more minority classes. In contrast, the proposed algorithms provide equitable performance across all classes, and are able to yield a much lower H-mean score. This demonstrates the advantage of using algorithms that directly optimize for the metric of interest. In most experiments, TFCO is seen to be a competitive baseline: with the H-mean metric, the proposed algorithms yields significantly better performance over this method on two of the five datasets, and with the micro F_1 metric it yields significantly better performance than TFCO on four of the five datasets . We stress that our algorithms are able to provide these gains despite TFCO using a more flexible class of randomized classifiers. In fact, with the MACHO dataset, TFCO can be seen to perform worse than our method as a result of over-fitting to the training set.

We also note that all the algorithms compared beat a trivial classifier that predicts all classes with equal probability (see Appendix B.2 for the performance of the trivial classifier on the different datasets with different metrics).

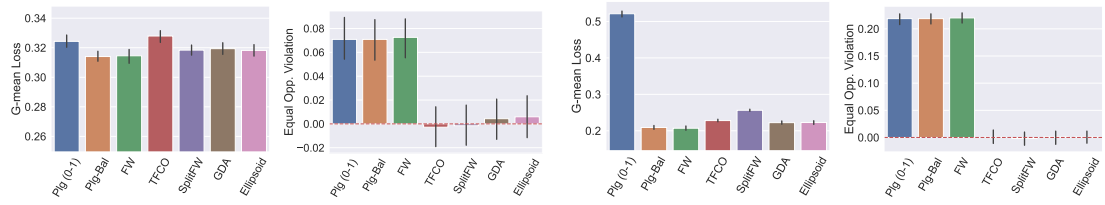
8.5 Performance on Constrained Problems

Having showed the efficacy of our algorithms on unconstrained problems, we move to constrained problems. The first task we consider is to minimize the H-mean loss subject to coverage constraint



(a) PageBlock (b) SatImage

Figure 8: Optimizing the H-mean loss subject to the coverage constraint $\max_i |\sum_j C_{ji} - \pi_i| \leq 0.01$. The plots on the left show the H-mean loss on the test set and those on the right show the coverage violation $\max_i |\sum_j C_{ji} - \pi_i| - 0.01$ on the test set. *Lower* H-mean value are *better*, and the constraint values need to be ≤ 0. The results are averaged over 10 random train-test splits. The error bars indicate 95% confidence intervals.

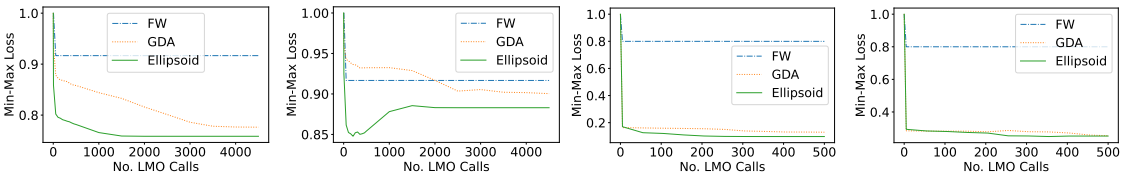


(a) COMPAS (b) Law School

Figure 9: Optimizing the G-mean loss subject to the equal-opportunity fairness constraint $\max_{a\in[m]} \left|\frac{1}{\mu_{a1}} C^a_{11} - \frac{1}{\pi_1} C_{11}\right| \leq 0.05$. The plots on the left show the G-mean loss on the test set and those on the right show the equal opportunity violation $\max_{a\in[m]} \left|\frac{1}{\mu_{a1}} C^a_{11} - \frac{1}{\pi_1} C_{11}\right|$ on the test set. *Lower* G-mean value are *better*, and the constraint violations need to be ≤ 0. The results are averaged over 10 random train-test splits. The error bars indicate 95% confidence intervals.

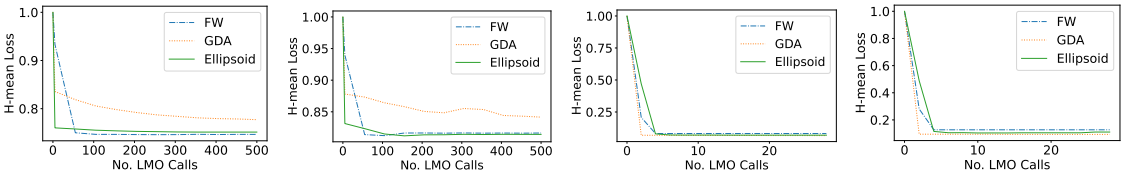
requiring the proportion of predictions for each class i to match the class prior π. Specifically, we require the maximum coverage violation over the n classes $\max_{i\in[n]} |\sum_j C_{ji} - \pi_i|$ to be at most 0.01. In Figure 8, we report both the H-mean and the maximum coverage violation for the three proposed constrained learning algorithms (Algorithms 5–7) for this problem (see Appendix B.3 for additional results). For comparison, we also report the performance of the 0-1 plug-in, balanced plug-in, and TFCO baselines, as well as the unconstrained Frank-Wolfe (FW) method, which seeks to optimize only the H-mean ignoring the constraint. We find that all three algorithms satisfy the constraint on the training set, but occasionally incur some violations on the test set. In contrast, all baselines expect TFCO fail to satisfy the constraint. On SatImage, TFCO satisfies the constraint on the training set, but fails to satisfy it on the test set, while the proposed methods incur much lower test violations. This is also the case with MACHO, where TFCO incurs lower constraint violation and loss value on the training set, but compared to our methods is worse of on both metrics on the test set. The reason our methods are less prone to over-fitting is because they use a plug-in based LMO that post-shifts a pre-trained class-probability estimator, and therefore have fewer parameters to optimize when compared to TFCO.

Our second task seeks to impose fairness constraints on benchmark fair classification datasets containing protected group information. These include: (1) *COMPAS*, where the goal is to predict recidivism with *gender* as the protected attribute (Angwin et al., 2016); (2) *Communities &*



(a) Abalone (Train) (b) Abalone (Test) (c) PageBlock (Train) (d) PageBlock (Test)

Figure 10: Optimizing the *Min-max* loss: Comparison of performance of the Frank-Wolfe, GDA and ellipsoid methods as a function of the number of LMO calls. *Lower* values are *better*. Because the min-max loss is non-smooth, Frank-Wolfe is seen to converge to a sub-optimal classifier.

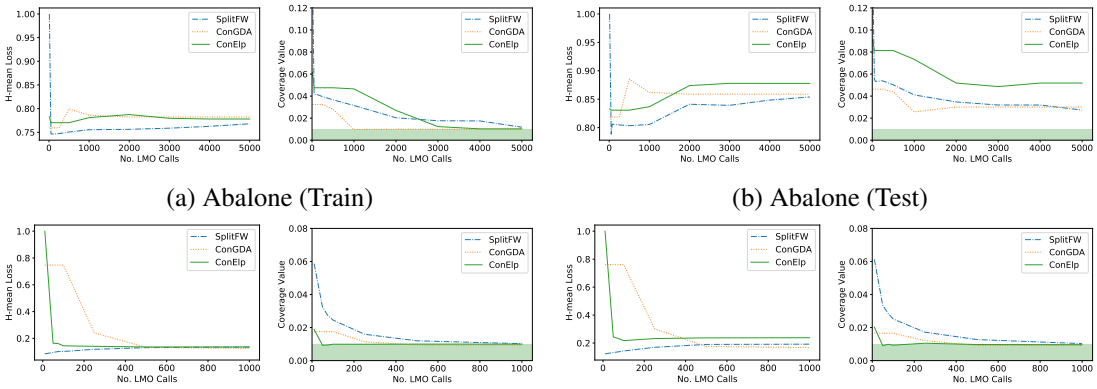


(a) Abalone (Train) (b) Abalone (Test) (c) PageBlock (Train) (d) PageBlock (Test)

Figure 11: Optimizing the *H-mean* loss: Comparison of performance of the Frank-Wolfe, GDA and ellipsoid methods as a function of the number of LMO calls. *Lower* values are *better*.

Crime, where the goal is to predict if a community in the US has a crime rate above the 70th percentile (Frank and Asuncion, 2010), and we consider communities having a black population above the 50th percentile as protected (Kearns et al., 2018); (3) *Law School*, where the task is to predict whether a law school student will pass the bar exam, with *race* (black or other) as the protected attribute (Wightman, 1998); (4) *Adult*, where the task is to predict if a person's income exceeds 50K/year, with *gender* as the protected attribute (Frank and Asuncion, 2010); (5) *Default*, where the task is to predict if a credit card user defaulted on a payment, with gender as the protected attribute (Frank and Asuncion, 2010). While these are all binary-labelled datasets, because we wish to evaluate performance separately on the individual protected groups, the number of threshold parameters needed to learn a naïve plug-in classifier like the one described in Section 3.3 would grow exponentially with the number of groups, making the algorithms proposed in this paper desirable even in these multi-group settings.

The specific optimization goal is to minimize the G-mean loss $\psi^{\text{GM}}(\mathbf{C}) = 1 - \left(\prod_i \frac{C_{ii}}{\sum_j C_{ij}}\right)^{1/n}$ subject to an equal opportunity constraint $\max_{a\in[m]} \left|\frac{1}{\mu_{a1}} C^a_{11} - \frac{1}{\pi_1} C_{11}\right| \leq 0.05$, requiring the true positive rates for different protected groups to be similar. The plots in Figure 9 presents the results for the three proposed algorithms relevant to this problem, and show both the G-mean loss and the equal opportunity violation (more results in Appendix B.3). In addition to the 0-1 plug-in, balanced plug-in and TFCO baselines, we include an unconstrained Frank-Wolfe (FW) method which seeks to minimize only the G-mean ignoring the constraint. All these methods incur large constrained violations. The objectives are largely comparable for the three proposed methods, except on LawSchool, where SplitFW yields a higher loss. The constraint violations for our methods are comparable to or lower than TFCO, with TFCO failing to satisfy the constraint on Crimes as a result of over-fitting to the training set.



(a) Abalone (Train)

(b) Abalone (Test)

(c) PageBlock (Train)

(d) PageBlock (Test)

Figure 12: Optimizing H-mean subject to coverage constraint: Comparison of performance of SplitFW, ConGDA and ConEllipsoid algorithms as a function of the number of LMO calls. *Lower* H-mean values are *better*. Green shaded region denotes coverage values that satisfy the constraints.

8.6 Practical Guidance on Algorithm Choice

Of the three types of algorithms we have proposed for convex metrics ψ, the choice of the algorithm to use in an application would depend on three factors: the smoothness of the metric, the presence of constraints, and the dimension of the problem. In Figure 10, we consider the task of optimizing the min-max metric $\psi^{\text{MM}}(\mathbf{C}) = \max_i \left(1 - \frac{C_{ii}}{\sum_j C_{ij}}\right)$, a *non-smooth* function of $\mathbf{C}$, and plot the performance of the three algorithms (with a plug-in based LMO) on the training and test sets as a function of the number of calls to the LMO. Since the objective for this unconstrained problem does not satisfy the smoothness property required by the Frank-Wolfe algorithm, as expected, it fairs poorly even with a large number of LMO calls. The ellipsoid algorithm is often seen to exhibit faster convergence than GDA on the training set, but there isn't a clear winner on the test set. In Figure 10, we repeat the experiment with the smooth H-mean metric, and find that Frank-Wolfe algorithm does converge to a similar performance as the other methods, and is in fact the fastest to do so on the 12-class Abalone dataset. Moreover unlike the GDA, the Frank-Wolfe algorithm has no additional hyper-parameters to tune and is therefore an attractive option for smooth convex metrics.

On the other hand, when it comes to constrained problems, we find the (constrained) GDA algorithm to exhibit the fastest convergence. In this case, the (constrained) ellipsoid algorithm may take longer to converge to the optimal-feasible solution, particularly when the number of classes is high (as seen from the strong dependence on dimension its convergence rate has in Theorem 18). For example, this evident with the 12-class Abalone dataset in Figure 12(a)–(b), where we seek to maximize the H-mean loss subject to the coverage constraint described in Section 8.5, and find the GDA algorithm to converge the fastest to a feasible classifier. In contrast, the (constrained) ellipsoid algorithm exhibits the fastest convergence on the smaller 5-class PageBlock dataset (although it yields slightly worse H-mean values than the other methods on the test set). See Appendix B.3 for additional experimental results.

Overall, we prescribe using the ellipsoid algorithm (or its constrained counterpart) for problems with a small number of classes, the FrankWolfe algorithm if the metric is smooth and there are no constraints, and the GDA algorithm (or its constrained counterpart) for all other scenarios.

Table 8: Comparison of the plug-in and weighted logistic regression (WLR) based LMOs on the task of optimizing the (convex) H-mean loss. The number of iterations, i.e. calls to the LMO, is fixed at *20*. *Lower* values are *better*. The results are averaged over 10 random train-test splits.

Data	FW		GDA		Ellipsoid	
	Plugin	**WLR**	**Plugin**	**WLR**	**Plugin**	**WLR**
Aba	0.797 ± 0.008	$\mathbf{0.791 \pm 0.004}$	0.892 ± 0.038	$\mathbf{0.838 \pm 0.017}$	0.833 ± 0.038	0.833 ± 0.038
PgB	0.13 ± 0.038	$\mathbf{0.084 \pm 0.015}$	0.129 ± 0.034	$\mathbf{0.083 \pm 0.018}$	0.105 ± 0.019	$\mathbf{0.080 \pm 0.017}$
MAC	$\mathbf{0.125 \pm 0.01}7$	0.245 ± 0.027	$\mathbf{0.124 \pm 0.015}$	0.206 ± 0.028	$\mathbf{0.122 \pm 0.015}$	0.247 ± 0.027
Sat	0.174 ± 0.007	$\mathbf{0.171 \pm 0.007}$	$\mathbf{0.173 \pm 0.008}$	0.176 ± 0.006	0.168 ± 0.006	$\mathbf{0.167 \pm 0.006}$
Cov	0.468 ± 0.001	$\mathbf{0.453 \pm 0.001}$	0.488 ± 0.001	$\mathbf{0.453 \pm 0.001}$	0.463 ± 0.001	$\mathbf{0.447 \pm 0.001}$

8.7 Choice of LMO: Plug-in vs. Weighted Logistic Regression

In previous experiments, we have seen that the proposed algorithms were less prone to over-fitting because of the use of a plug-in based LMO that post-fit a small number of parameters to a pre-trained model. We now compare the performance of these algorithms with an LMO that re-trains a classifier from scratch each time it is called. Specifically, we repeat the H-mean optimization task from Section 8.4, with weighted logistic regression on a linear model as the LMO. For a given (diagonal) loss matrix $\mathbf{L}$, this LMO learns a classifier by optimizing a weighted logistic loss, where the per-class weights are set to be the diagonal entries of $\mathbf{L}$. Note that such a weighted surrogate loss is calibrated for $\mathbf{L}$ (Tewari and Bartlett, 2007). Unlike the simple plug-in LMO, each call to weighted logistic regression can be expensive; hence it is important that we are able to limit the number of calls to it.

In Table 8, we present results comparing performance of the Frank-Wolfe, GDA and ellipsoid algorithms with the plug-in and weighted logistic regression LMOs when run for 20 iterations. Appendix B.3 contains results of these experiments when the algorithms are allowed 100 iterations. The performance with the two LMOs are comparable on Abalone and SatImage. On PageBlocks and CovType, weighted logistic regression has a moderate to significant advantage. Interestingly, on MACHO, the plug-in based LMO, despite learning from a less flexible hypothesis class (post-hoc adjustments to a fixed model), is substantially better. This is because weighted logistic regression over-fits to the training set in this case.

Overall, we find that an LMO such as weighted logistic regression, while being computationally expensive, does sometimes provide metric gains over a less-flexible plug-in type approach. However, this method can be prone to over-fitting because of its added flexibility.

8.8 Case Study: Image Classification with Imbalance and Label Noise

As case studies, we demonstrate two natural workflows our algorithms in (i) tackling label imbalance in CIFAR-55 and (ii) mitigating label noise in a noisy version of CIFAR-10.

8.8.1 CLASS IMBALANCE WITH LARGE NUMBER OF CLASSES

One of the undesirable effects of learning with a class-imbalanced dataset is that the learned classifier tends to over-predict classes that are more prevalent and under-predict classes that are rare. We consider two approaches to avoid this problem: minimizing a loss such as the H-mean that emphasizes equal performance across all classes, and constraining the proportions of predictions the classifier makes for each class to match the true prevalence of the class.

Table 9: Results on CIFAR-55 imbalanced dataset. The train and test sets are imbalanced, with 5 classes being 10 times larger in size than the remaining 50 classes. We report the 0-1 loss, the H-mean loss, and the coverage violation $\max_{i\in[n]} |\sum_j C_{ji} - \pi_i| - 0.01$. *Lower* values are *better*.

Method	Train (Imbalanced)			Test (Imbalanced)		
	0-1	H-mean	Violation	0-1	H-mean	Violation
Plugin [0-1]	**0.278**	0.457	0.030	0.437	0.709	0.045
FW [H-mean]	0.307	**0.323**	0.026	0.481	**0.564**	0.029
SplitFW [0-1]	0.279	0.391	**0.000**	**0.436**	0.636	0.007
SplitFW [H-mean]	0.279	0.342	**0.000**	0.448	0.595	**0.000**

For this experiment, we use the CIFAR-100 dataset (Krizhevsky, 2009), which contains images labelled with one of 100 classes. We create an imbalanced 55-class dataset by merging 50 classes in CIFAR-100 into 5 "super-classes" (see Appendix B.2 for details), and leaving the rest of the classes untouched. In the resulting class distribution, 5 of the classes are 10 times more prevalent than the remaining 50. All our methods use a plug-in based LMO which uses a pre-trained class probability estimator. We train a ResNet-50 model for the class probability estimator $\widehat{\boldsymbol{\eta}}$, using SGD to minimize the standard cross-entropy loss. We use a batch size of 64, a base learning rate of 0.01 (with a warm-up cosine schedule), and a momentum of 0.9. We apply a weight decay of 0.01 and train for 39 epochs.

In Table 9, we analyze the performance of a ResNet-50 model trained with the standard cross-entropy loss (Plugin [0-1]), where the class that receives the highest estimated probability is predicted as the output label, and report its 0-1 loss, its H-mean loss, and the deviation of its class prediction rates from the prior probabilities, i.e. its maximum coverage violation: $\max_{i\in[n]} |\sum_j C_{ji} - \pi_i|$. We find that naïvely optimizing for the 0-1 loss yields a high coverage violation. Moreover, it yields high accuracies on the 5 super-classes, at the cost of a much lower accuracy on the 50 minority classes, resulting in a high H-mean loss. To emphasize better performance on the minority classes, we train classifiers to minimize the H-mean loss (FW [H-mean]), and minimize the 0-1 loss subject to the maximum coverage violation being within a tolerance of 0.01 (SplitFW [0-1]). We also consider a combination of both, i.e. minimizing the H-mean loss subject to the maximum coverage violation being within 0.01 (SplitFW [H-mean]). It can be seen that all three algorithms do only slightly worse than the Plugin [0-1] baseline in terms of 0-1 loss, but do significantly better in terms of both the H-mean loss and the coverage violation.

8.8.2 CLASS IMBALANCE WITH LABEL NOISE

Our next experiment demonstrates how label noise can be mitigating by imposing coverage constraints on the classifier. We use a class-imbalanced version of the CIFAR-10 dataset (Krizhevsky, 2009), where we sub-sample images from classes 1 to 5 by a factor of 10, with the resulting class priors are given by $\pi_y = \frac{2}{110}$ when $y \in \{1, 2, 3, 4, 5\}$ and $\pi_y = \frac{2}{11}$ otherwise. Our algorithms assume the knowledge of π. In addition to class imbalance, very often one has to work with noisy training labels to building a classifier that performs well on uncorrupted test data. We simulate this scenario by adding a label noise corruption to the training data, which is chosen such that classes 1 to 9 are left undisturbed, and the labels of images from class 10 are randomly chosen from 1 to 10. This effectively mimics a crowd-sourced label collection with 9 easy labels, and one difficult or incomprehensible label. Our algorithms do *not* have knowledge of this corruption.

Table 10: Results on imbalanced CIFAR-10 dataset with label noise. The train set is imbalanced and has label noise, while the test set is imbalanced but clean. We report the 0-1 loss, the H-mean loss, and the coverage violation $\max_{i\in[n]} |\sum_j C_{ji} - \pi_i| - 0.01$. *Lower* values are *better*.

Method	Train (Flipped)			Test (Imbalanced)		
	0-1	**H-mean**	**Violation**	**0-1**	**H-mean**	**Violation**
Plugin [0-1]	0.266	0.896	0.170	0.332	0.899	0.169
Noise Correction [Estimate]	**0.174**	**0.370**	0.054	**0.179**	0.430	0.055
FW [H-mean]	0.348	0.481	0.072	0.329	**0.396**	0.076
SplitFW [0-1]	0.272	0.609	**0.001**	0.196	0.610	0.004
SplitFW [H-mean]	0.292	0.523	0.003	0.221	0.471	**0.003**
Noise Correction [Exact]	0.196	0.394	0.022	0.151	0.358	0.018

Equipped with the knowledge of the class priors π, we propose constraining the proportion of predictions made for each class to match the priors π. While this is not necessarily equivalent to training the classifier with uncorrupted labels, we expect that these additional coverage constraints will dampen the effect of the noisy labels. As with the previous experiment, we evaluate the classifier on two criteria: (i) how well it performs on the (balanced) H-mean metric on the test data, and (ii) how well the class prediction rates match the priors π on the test data. Our framework can be applied to this problem by minimizing the H-mean error on the corrupted training dataset, subject to a coverage constraint on the classifier forcing it to predict classes at a rate that matches π.

In addition to a ResNet model baseline that optimizes the cross-entropy loss (Plugin [0-1]), we include the state-of-the-art method of Patrini et al. (2017), which uses the predictions from Plugin [0-1] on the training data to compute an estimate of the label noise transition matrix, and re-trains the classifier with a (forward) correction computation applied to the loss (Noise Correction [Estimate]). For completeness, we also include an *idealized* version of this method, where the "exact" noise transition matrix is used for the forward correction (Noise Correction [Exact]). While this baseline is unrealistic, it provides us with an estimate best possible 0-1 loss achievable for this problem.

We provide the result of this experiment in Table 10, where FW [H-mean] corresponds to a classifier that minimizes the H-mean loss on the corrupted training dataset, and SplitFW [0-1] (resp. SplitFW [H-mean]), correspond to a classifier that minimizes the 0-1 loss (resp. H-mean loss) on the corrupted training dataset, while enforcing the coverage constraint to a tolerance of 0.01. All three methods use the same underlying class probability model as Plugin [0-1]. It is seen that only SplitFW [0-1] and SplitFW [H-mean] achieve low coverage violations on the test set, and are still only moderately worse than the idealized Noise Correction [Exact] method in terms of their respective objective metrics. The FW [H-mean] algorithm achieves the best H-mean on clean test data despite being trained on the corrupted training labels.

9. Conclusions

We have developed a framework for designing consistent and efficient algorithms for multiclass performance metrics and constraints that are general functions of the confusion matrix. As instantiations of this framework, we provided four algorithms for optimizing unconstrained metrics, and four analogous counterparts for solving constrained learning problems. In each case, we have shown convergence guarantees for the algorithms under different assumptions on the performance metrics and constraints.

Our key idea was to reduce the complex learning problem into a sequence of linear minimization problems, for which we recommended an efficient plug-in based approach that applies a post-hoc transformations to a pre-trained class probability model. The results of these linear minimization problems are then combined to return a final classifier. One of the main challenges in instantiating this idea was to identify optimization algorithms for different problem settings that only required access to a linear minimization oracle (LMO).

We also presented extensive experiments on a variety of multiclass and fairness datasets and demonstrated that the proposed algorithms (despite being limited to performing adjustments to a fixed model) are competitive or better than the state-of-the-art TFCO approach (Cotter et al., 2019b) which works with a more flexible hypothesis class. We additionally provided precise guidance for which of the proposed algorithms are best suited for a given multiclass problem, and highlighted scenarios where one might want to use a more expensive LMO that trains a new classifier from scratch at each iteration.

Over the years, the conference versions of this paper have attracted several follow-up works, which have adapted our ideas to optimizing multiclass extensions of the F-measure (Pan et al., 2016), to balancing accuracy with fairness objectives (Alabi et al., 2018), to eliciting multi-class performance metrics (Hiranandani et al., 2019), to training classifiers to optimize more general multi-output classification metrics (Wang et al., 2019), to imposing fairness constraints with overlapping protected groups (Yang et al., 2020), and to optimizing black-box evaluation metrics Hiranandani et al. (2021).

A number of follow-up directions arise from the proposed framework. First, it would interesting to derive lower bounds on the number of calls to the LMO needed under different assumptions on the performance metrics and constraints.

Second, while the optimality (and feasibility) gap for most of our proposed algorithms depend linearly on the LMO approximation errors ρ and ρ', the split Frank-Wolfe method (Algorithm 5 alone has a square-root dependence on these parameters. Are these dependencies on the LMO errors optimal or simply artifacts of the analysis?

Third, for algorithms where the convergence rates have a linear (or quadratic) dependence on the dimension of the problem d (which is typically the same order as the number of classes), how does one extend our framework to handle problems with an extremely large number classes (perhaps under additional structural assumptions on the classes, akin to Ramaswamy et al. (2015)) and problems with an extremely large number of constraints (Narasimhan et al., 2020)?

Fourth, our experiments in Section 8.7 show that in some applications, using a flexible LMO that trains a classifier from scratch can yield significant gains over a plug-in based LMO, but this however comes at the cost of added computational time. Can one devise an intermediate approach, where each call to the LMO only needs to run a constant number of optimization steps on a surrogate loss (akin to the TFCO baseline of Cotter et al. (2019b)), while still guaranteeing that the outer algorithm provably convergences to the optimal (feasible) classifier?

Finally, except for the bisection method, all the algorithms we propose rely on the use of a randomized classifier. In some applications, deploying a randomized classifier can be undesirable for ethical reasons or because of the engineering difficulties it poses. In these scenarios, one could approximate the learned randomized classifier with a deterministic classifier using, for example, the approach of Cotter et al. (2019). Understanding the loss in performance and constraint satisfaction as a result of such de-randomization procedures is an interesting direction for future work.

Acknowledgements

The authors thank Aadirupa Saha for providing helpful inputs and for running experiments for a conference version of this paper (Narasimhan et al., 2015b). HG thanks the Robert Bosch Centre for Data Science and Artificial Intelligence for their support. HN thanks Pavlos Protopapas, IACS, Harvard University, for providing us access to the MACHO celestial object detection dataset.

References

J. D. Abernethy and J.-K. Wang. On Frank-Wolfe and equilibrium computation. In *NIPS*, 2017.

A. Agarwal, A. Beygelzimer, M. Dudik, J. Langford, and H. Wallach. A reductions approach to fair classification. In *ICML*, 2018.

D. Alabi, N. Immorlica, and A. Kalai. Unleashing linear optimizers for group-fair learning and optimization. In *COLT*, 2018.

C. Alcock, R. A. Allsman, D. R. Alves, T. S. Axelrod, A. C. Becker, D. P. Bennett, K. H. Cook, N. Dalal, A. J. Drake, K. C. Freeman, et al. The MACHO project: Microlensing results from 5.7 years of large magellanic cloud observations. *The Astrophysical Journal*, 542(1):281, 2000.

J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias. *ProPublica, May*, 23, 2016.

P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Alexander Barvinok. *A course in convexity*, volume 54. American Mathematical Soc., 2002.

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36:929–965, 1989.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8:231–358, 2015.

C. Calauzènes, N. Usunier, and P. Gallinari. On the (non-)existence of convex, calibrated surrogate losses for ranking. In *NIPS*, 2012.

L. E. Celis, L. Huang, V. Keswani, and N. K. Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *FAT*, 2019.

R. Chen, B. Lucier, Y. Singer, and V. Syrgkanis. Robust optimization for non-convex objectives. *arXiv preprint arXiv:1707.01047*, 2017.

Q. Cormier, M.M. Fard, K. Canini, and M. Gupta. Launch and iterate: Reducing prediction churn. In *NIPS*, 2016.

A. Cotter, H. Jiang, and K. Sridharan. Two-player games for efficient non-convex constrained optimization. In *ALT*, 2019a.

A. Cotter, H. Jiang, S. Wang, T. Narayan, M. Gupta, S. You, and K. Sridharan. Optimization with non-differentiable constraints with applications to fairness, recall, churn, and other goals. *Journal of Machine Learning Research*, 2019b.

A. Cotter, H. Narasimhan, and M. Gupta. On making stochastic classifiers deterministic. In *NeurIPS*, 2019.

J. M. Danskin. *The theory of max-min and its application to weapons allocation problems*, volume 5. Springer Science & Business Media, 2012.

K. Dembczynski, W. Waegeman, W. Cheng, and E. Hüllermeier. An exact algorithm for F-measure maximization. In *NIPS*, 2011.

K. Dembczynski, A. Jachnik, W. Kotlowski, W. Waegeman, and E. Hullermeier. Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In *ICML*, 2013.

K. Dembczyński, W. Kotłowski, O. Koyejo, and N. Natarajan. Consistency analysis for binary classification revisited. In *ICML*, 2017.

M. Donini, L. Oneto, S. Ben-David, J.S. Shawe-Taylor, and M. Pontil. Empirical risk minimization under fairness constraints. In *NeurIPS*, 2018.

J. Duchi, L. Mackey, and M. Jordan. On the consistency of ranking algorithms. In *ICML*, 2010.

E. Eban, M. Schain, A. Mackey, A. Gordon, R. Rifkin, and G. Elidan. Scalable learning of non-decomposable objectives. In *AISTATS*, 2017.

C. Elkan. The foundations of cost-sensitive learning. In *IJCAI*, 2001.

A. Esuli and F. Sebastiani. Optimizing text quantifiers for multivariate loss functions. *ACM Transactions on Knowledge Discovery and Data*, 9(4):Article 27, 2015.

J. Finocchiaro, R. M. Frongillo, and B. Waggoner. Embedding dimension of polyhedral losses. In *COLT*, 2020.

A. Frank and A. Asuncion. UCI machine learning repository. URL: `http://archive.ics.uci.edu/ml`, 2010.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.

W. Gao and F. Sebastiani. Tweet sentiment: From classification to quantification. In *ASONAM*, 2015.

W. Gao and Z.-H. Zhou. On the consistency of multi-label learning. In *COLT*, 2011.

G. Gidel, F. Pedregosa, and S. Lacoste-Julien. Frank-wolfe splitting via augmented lagrangian method. In *AISTATS*, 2018.

G. Goh, A. Cotter, M. Gupta, and M.P. Friedlander. Satisfying real-world goals with dataset constraints. In *NIPS*, 2016.

M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *NIPS*, 2016.

G. Hiranandani, S. Boodaghians, R. Mehta, and O. Koyejo. Multiclass performance metric elicitation. *NeurIPS*, 2019.

G. Hiranandani, J. Mathur, H. Narasimhan, M. M. Fard, and O. Koyejo. Optimizing black-box metrics with iterative example weighting. In *ICML*, 2021.

M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML*, 2013.

T. Joachims. A support vector method for multivariate performance measures. In *ICML*, 2005.

P. Kar, H. Narasimhan, and P. Jain. Online and stochastic gradient methods for non-decomposable loss functions. In *NeurIPS*, 2014.

P. Kar, S. Li, H. Narasimhan, S. Chawla, and F. Sebastiani. Online optimization methods for the quantification problem. In *KDD*, 2016.

M. Kearns, S. Neel, A. Roth, and Z.S. Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *ICML*, 2018.

K. Kennedy, B. M. Namee, and S. J. Delany. Learning without default: A study of one-class classification and the low-default portfolio problem. In *ICAICS*, 2009.

D.-W. Kim, P. Protopapas, Y.-I. Byun, C. Alcock, R. Khardon, and M. Trichas. Quasi-stellar object selection algorithm using time variability and machine learning: Selection of 1620 quasi-stellar object candidates from macho large magellanic cloud database. *The Astrophysical Journal*, 735 (2):68, 2011.

J-D. Kim, Y. Wang, and Y. Yasunori. The genia event extraction shared task, 2013 edition - overview. *ACL*, 2013.

O. Koyejo, N. Natarajan, P. Ravikumar, and I.S. Dhillon. Consistent binary classification with generalized performance metrics. In *NIPS*, 2014.

A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

A. Kumar, H. Narasimhan, and A. Cotter. Implicit rate-constrained optimization of non-decomposable objectives. In *ICML*, 2021.

S. Lawrence, I. Burns, A. Back, A-C. Tsoi, and C.L. Giles. Neural network classification and prior class probabilities. In *Neural Networks: Tricks of the Trade*, LNCS, pages 1524:299–313. 1998.

Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines: Theory and application to the classification of microarray data. *Journal of the American Statistical Association*, 99(465): 67–81, 2004.

Y. T. Lee, A. Sidford, and S. C. Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *FOCS*, 2015.

D.D. Lewis. Evaluating text categorization. In *Proceedings of the Workshop on Speech and Natural Language*, HLT, 1991.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

A.K. Menon, H. Narasimhan, S. Agarwal, and S. Chawla. On the statistical consistency of algorithms for binary classification under class imbalance. In *ICML*, 2013.

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

H. Narasimhan, R. Vaish, and S. Agarwal. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In *NIPS*, 2014.

H. Narasimhan, P. Kar, and P. Jain. Optimizing non-decomposable performance measures: A tale of two classes. In *ICML*, 2015a.

H. Narasimhan, H.G. Ramaswamy, A. Saha, and S. Agarwal. Consistent multiclass algorithms for complex performance measures. In *ICML*, 2015b.

H. Narasimhan, A. Cotter, Y. Zhou, S. Wang, and W. Guo. Approximate heavily-constrained learning with lagrange multiplier models. *NeurIPS*, 2020.

H. Narasimhan, A. Cotter, and M. Gupta. Optimizing generalized rate metrics with three players. In *NeurIPS*, 2019.

N. Natarajan, O. Koyejo, P. Ravikumar, and I. Dhillon. Optimal classification with multivariate losses. In *ICML*, 2016.

A. Nowak-Vila, F. Bach, and A. Rudi. Consistent structured prediction with max-min margin markov networks. In *ICML*, 2020.

W. Pan, H. Narasimhan, P. Kar, P. Protopapas, and H. G. Ramaswamy. Optimizing the multiclass F-measure via biconcave programming. In *ICDM*, 2016.

S.A.P. Parambath, N. Usunier, and Y. Grandvalet. Optimizing F-measures by cost-sensitive classification. In *NIPS*, 2014.

G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, 2017.

B. Á. Pires, C. Szepesvari, and M. Ghavamzadeh. Cost-sensitive multiclass classification risk bounds. In *ICML*, 2013.

H. Ramaswamy, A. Tewari, and S. Agarwal. Convex calibrated surrogates for hierarchical classification. In *ICML*, 2015.

H. G. Ramaswamy and S. Agarwal. Classification calibration dimension for general multiclass losses. In *NIPS*, 2012.

H. G. Ramaswamy, S. Agarwal, and A. Tewari. Convex calibrated surrogates for low-rank loss matrices with applications to subset ranking losses. In *NIPS*, 2013.

H. G. Ramaswamy, A. Tewari, and S. Agarwal. Consistent algorithms for multiclass classification with an abstain option. *Electronic Journal of Statistics*, 12(1):530–554, 2018.

P. Ravikumar, A. Tewari, and E. Yang. On NDCG consistency of listwise ranking methods. In *AISTATS*, 2011.

A. Sanyal, P. Kumar, P. Kar, S. Chawla, and F. Sebastiani. Optimizing non-decomposable measures with deep networks. *Machine Learning*, 107(8-10):1597–1620, 2018.

S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2):107–194, 2011.

I. Steinwart. How to compare different loss functions and their risks. *Constructive Approximation*, 26:225–287, 2007.

Y. Sun, M.S. Kamel, and Y. Wang. Boosting for learning multiple classes with imbalanced class distribution. In *ICDM*, 2006.

S. K. Tavker, H. G. Ramaswamy, and H. Narasimhan. Consistent plug-in classifiers for complex objectives and constraints. In *NeurIPS*, 2020.

A. Tewari and P. L. Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8:1007–1025, 2007.

K. M. Ting. An instance-weighting method to induce cost-sensitive trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(3):659–665, 2002.

E. Vernet, R. C. Williamson, and M. D. Reid. Composite multiclass losses. In *NIPS*, 2011.

P. H. Vincent. *An Introduction to Signal Detection and Estimation*. Springer-Verlag, 1994.

Willem Waegeman, Krzysztof Dembczyński, Arkadiusz Jachnik, Weiwei Cheng, and Eyke Hüllermeier. On the bayes-optimality of f-measure maximizers. *Journal of Machine Learning Research*, 15:3333–3388, 2014.

S. Wang and X. Yao. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(4):1119–1130, 2012.

X. Wang, R. Li, B. Yan, and O. Koyejo. Consistent classification with generalized metrics. *arXiv preprint arXiv:1908.09057*, 2019.

L. Wightman. Lsac national longitudinal bar passage study. *Law School Admission Council*, 1998.

F. Yang and O. Koyejo. On the consistency of top-k surrogate losses. In *ICML*, 2020.

F. Yang, M. Cisse, and O. Koyejo. Fairness with overlapping groups; a probabilistic perspective. *NeurIPS*, 2020.

N. Ye, K. M. A. Chai, W. S. Lee, and H. L. Chieu. Optimizing F-measures: A tale of two approaches. In *ICML*, 2012.

Table 11: Table of notations.

Notation	Description
n	Number of classes
N	Number of training examples
m	Number of protected groups
K	Number of constraints
d	Dimension of the vector representation for the confusion matrix
T	Number of iterations for the proposed iterative algorithms
i, j	Indices over n classes
a	Index over m protected groups
k	Index over K constraints
t	Index over T iterations
ℓ	Index over N training instances
$\mathbf{C}$	$n \times n$ Confusion matrix, or an equivalent vector representation of dimension $d = n^2$
$\mathbf{L}$	$n \times n$ Loss matrix, or an equivalent vector representation of dimension $d = n^2$
$\mathcal{C}$	Set of achievable confusion matrices, represented by vectors of dimension $d = n^2$
S	Training sample with N instances

M. Yuan and M. Wegkamp. Classification methods with reject option based on convex risk minimization. *Journal of Machine Learning Research*, 11:111–130, 2010.

M. B. Zafar, I. Valera, M. G. Rodriguez, and K. P. Gummadi. Fairness constraints: Mechanisms for fair classification. In *AISTATS*, 2017a.

M. B. Zafar, I. Valera, M. G. Rodriguez, and K. P. Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *WWW*, 2017b.

M. Zhang, H. G. Ramaswamy, and S. Agarwal. Convex calibrated surrogates for the multi-label f-measure. In *ICML*, 2020.

T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32(1):56–134, 2004a.

T. Zhang. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research*, 5:1225–1251, 2004b.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, 2003.

Appendix A. Proofs

A.1 Proof of Proposition 8 (Bayes optimal Classifier for Ratio-of-linear ψ)

Proposition ((Restated) Bayes optimal classifier for ratio-of-linear ψ)**.** *Let the performance measure* $\psi : [0,1]^d \to \mathbb{R}_+$ *in* OP1 *be of the form* $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$ *for some* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^d$ *with* $\langle \mathbf{B}, \mathbf{C} \rangle > 0,\ \forall \mathbf{C} \in \mathcal{C}$*. Let* $t^* = \inf_{\mathbf{C} \in \mathcal{C}} \psi(\mathbf{C})$ *and* $\mathbf{L}^* = \mathbf{A} - t^* \mathbf{B}$*. Then any (deterministic) classifier that is optimal for the linear metric* $\langle \mathbf{L}^*, \mathbf{C} \rangle$ *is also optimal for* OP1.

We will find the following lemma useful in the proof of the proposition.

Lemma 25. *Let $\psi : [0,1]^d \to \mathbb{R}_+$ be such that $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$, for some matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^d$ with $\langle \mathbf{B}, \mathbf{C} \rangle > 0$ for all $\mathbf{C} \in \mathcal{C}$. Let $t^* = \inf_{\mathbf{C} \in \mathcal{C}} \psi(\mathbf{C})$. Then $\inf_{\mathbf{C} \in \mathcal{C}} \langle \mathbf{A} - t^* \mathbf{B}, \mathbf{C} \rangle = 0$.*

Proof. Define $\varphi : \mathbb{R} \to \mathbb{R}$ as $\varphi(t) = \inf_{\mathbf{C} \in \mathcal{C}} \langle \mathbf{A} - t\mathbf{B}, \mathbf{C} \rangle$. It is easy to see that φ (being a pointwise infimum of linear functions) is concave, and hence a continuous function over $\mathbb{R}$. Let $t^* = \inf_{\mathbf{C} \in \mathcal{C}} \psi(\mathbf{C})$. We then have for all $\mathbf{C} \in \mathcal{C}$,

$$\frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle} \geq t^* \quad \text{or equivalently} \quad \langle \mathbf{A} - t^* \mathbf{B}, \mathbf{C} \rangle \geq 0.$$

Thus

$$\varphi(t^*) = \inf_{\mathbf{C} \in \mathcal{C}} \langle \mathbf{A} - t^* \mathbf{B}, \mathbf{C} \rangle \geq 0 \,. \tag{10}$$

Also, by continuity of $\frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$ in $\mathbf{C}$, for any $t > t^*$, there exists $\mathbf{C} \in \mathcal{C}$ such that

$$\frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle} < t \quad \text{or equivalently} \quad \langle \mathbf{A} - t\mathbf{B}, \mathbf{C} \rangle < 0.$$

Thus for all $t > t^*$,

$$\varphi(t) = \inf_{\mathbf{C} \in \mathcal{C}} \langle \mathbf{A} - t\mathbf{B}, \mathbf{C} \rangle < 0 \,.$$

Next, by continuity of φ, for any monotonically decreasing sequence of real numbers $\{t_i\}_{i=1}^{\infty}$ converging to t^*, we have that $\varphi(t_i)$ converges to $\varphi(t^*)$. Since for each t_i in this sequence $\varphi(t_i) < 0$, and (10) states that $\varphi(t^*) \geq 0$, we have from the continuity of $\varphi(t)$ that:

$$\inf_{\mathbf{C} \in \mathcal{C}} \langle \mathbf{A} - t^* \mathbf{B}, \mathbf{C} \rangle = \varphi(t^*) = 0,$$

as desired. □

We are now ready to prove Proposition 8.

Proof of Proposition 8. Let $h^* : \mathcal{X} \to \Delta_n$ be a classifier that is optimal for $\mathbf{L}^* = \mathbf{A} - t^* \mathbf{B}$, i.e.,

$$\langle \mathbf{A} - t^* \mathbf{B}, \mathbf{C}[h^*] \rangle = \inf_{\mathbf{C} \in \mathcal{C}} \langle \mathbf{A} - t^* \mathbf{B}, \mathbf{C} \rangle.$$

Note from Lemma 25 that $\langle \mathbf{A} - t^* \mathbf{B}, \mathbf{C}[h^*] \rangle = 0$. Hence,

$$\frac{\langle \mathbf{A}, \mathbf{C}[h^*] \rangle}{\langle \mathbf{B}, \mathbf{C}[h^*] \rangle} = t^*, \text{or equivalently,} \quad \psi(\mathbf{C}[h^*]) = \inf_{\mathbf{C} \in \mathcal{C}} \psi(\mathbf{C}),$$

which shows that h^* is also ψ-optimal. Furthermore, from Proposition 5, h^* is also deterministic. □

A.2 Proof of Proposition 9 (Bayes optimal Classifier for Monotonic ψ)

Proposition ((Restated) Bayes optimal classifier for monotonic ψ)**.** *Let the performance measure $\psi : [0,1]^d \to \mathbb{R}_+$ in* OP1 *be differentiable and bounded, and be strictly decreasing in C_{ii} for each i and non-decreasing in C_{ij} for all $i \neq j$. Let $\boldsymbol{\eta}(X)$ be a continuous random vector. Then there exists a loss matrix $\mathbf{L}^*$ (which depends on ψ and D) such that any (deterministic) classifier that is optimal for the linear metric given by $\mathbf{L}^*$ is also optimal for* OP1.

Let $\overline{\mathcal{C}}$ denote the closure of $\mathcal{C}$. We will find the following lemma crucial to our proof.

Lemma 26. *Let $\boldsymbol{\eta}(X)$ be a continuous random vector. Let $\mathbf{L} \in \mathbb{R}^d$ be such that no two columns are identical. Then,*

$$\operatorname{argmin}_{\mathbf{C} \in \overline{\mathcal{C}}} \langle \mathbf{L}, \mathbf{C} \rangle \;=\; \operatorname{argmin}_{\mathbf{C} \in \mathcal{C}} \langle \mathbf{L}, \mathbf{C} \rangle.$$

Moreover, the above set is a singleton.

The proof for the lemma is highly technical and can be found in the conference version of the paper, specifically Lemma 12 in Narasimhan et al. (2015b). The following is a proof sketch. For any $\mathbf{L}$ with distinct columns, the optimal classifier given in Proposition 5 is uniquely defined except for a set of inputs $x \in \mathcal{X}$ where $\boldsymbol{\eta}(x)$ takes values in a $n-2$ dimensional manifold that is a subset of Δ_n. As $\boldsymbol{\eta}(X)$ is assumed to be a continuous random vector, this set of inputs for which the optimal classifier is not uniquely defined has probability 0. Since classifiers that have the same output for all but a zero probability fraction of the inputs have the same confusion matrix, the minimizer of $\langle \mathbf{L}, \mathbf{C} \rangle$ over $\mathbf{C} \in \mathcal{C}$ exists and is unique. Also, expanding the set $\mathcal{C}$ to $\overline{\mathcal{C}}$ does not give a better solution.

Proof of Proposition 9. Let $\mathbf{C}^* = \operatorname{argmin}_{\mathbf{C} \in \overline{\mathcal{C}}} \psi(\mathbf{C})$. Such a $\mathbf{C}^*$ always exists by compactness of $\overline{\mathcal{C}}$ and continuity of ψ. By first order optimality, and convexity of $\overline{\mathcal{C}}$, we have that for all $\mathbf{C} \in \overline{\mathcal{C}}$

$$\langle \nabla \psi(\mathbf{C}^*), \mathbf{C}^* \rangle \leq \langle \nabla \psi(\mathbf{C}^*), \mathbf{C} \rangle \; .$$

For $\mathbf{L}^* = \nabla \psi(\mathbf{C}^*)$, we have that $\mathbf{C}^* \in \operatorname{argmin}_{\mathbf{C} \in \overline{\mathcal{C}}} \langle \mathbf{L}^*, \mathbf{C} \rangle$.

Due to the strict decreasing condition on the diagonal elements of ψ, its gradient $\nabla \psi(\mathbf{C}^*)$ are negative, and the off-diagonal elements are non-negative, and hence no two columns of $\mathbf{L}^*$ are identical. Thus by a direct application of Lemma 26, we have that $\mathbf{C}^* \in \mathcal{C}$, and moreover $\mathbf{C}^*$ is the unique minimizer of $\langle \mathbf{L}^*, \mathbf{C} \rangle$ over all $\mathbf{C} \in \mathcal{C}$. From Proposition 5, we have that this minimizer is achieved by a deterministic classifier. □

A.3 Proof of Proposition 10 (Bayes optimal Classifier for Continuous $\psi, \phi_1, \ldots, \phi_K$)

Proposition (Bayes optimal classifier for continuous $\psi, \phi_1, \ldots, \phi_K$)**.** *Let the performance measure $\psi : [0,1]^d \to \mathbb{R}_+$ and the constraint functions $\phi_1, \ldots, \phi_K : [0,1]^d \to \mathbb{R}$ in* OP2 *be continuous and bounded. Then there exists $d+1$ loss matrices $\mathbf{L}_1^*, \mathbf{L}_2^*, \ldots, \mathbf{L}_{d+1}^*$ (which can depend on ψ, ϕ_k's and D) such that an optimal classifier for* OP2 *can be expressed as a randomized combination of the deterministic classifiers $h_1, h_2, \ldots, h_{d+1}$, where h_i is any optimal classifier for the linear metric given by $\mathbf{L}_i^*$.*

Proof. We first define the set feasible confusion matrices, and the set of achievable confusion matrices that are also feasible:

$$\mathcal{A} = \cap_{j=1}^{K} \{ \mathbf{C} \in [0,1]^d : \phi_j(\mathbf{C}) \leq 0 \}$$
$$\mathcal{B} = \mathcal{A} \cap \mathcal{C},$$

Let $\overline{\mathcal{B}}$ and $\overline{\mathcal{C}}$ be the closure of $\mathcal{B}$ and $\mathcal{C}$ respectively. As the functions ϕ_j are continuous, the set $\mathcal{A}$ is closed. Hence, $\overline{\mathcal{B}} = \mathcal{A} \cap \overline{\mathcal{C}}$. Let $\mathbf{C}^* \in \operatorname{argmin}_{\mathbf{C} \in \overline{\mathcal{B}}} \psi(\mathbf{C})$. Clearly such a $\mathbf{C}^*$ exists because ψ is continuous and $\overline{\mathcal{B}}$ is closed.

To complete the proof, it is sufficient to show that $\mathbf{C}^* \in \mathcal{B}$ and is equal to the confusion matrix of a classifier obtained by a randomized combination of $d+1$ deterministic classifiers. We already have that $\mathbf{C}^* \in \overline{\mathcal{C}}$ and $\mathbf{C}^* \in \mathcal{A}$. So we just need to show $\mathbf{C}^* \in \mathcal{C}$ and is equal to the confusion matrix of a classifier obtained by a randomized combination of $d+1$ deterministic classifiers.

By the Krein-Millman theorem, we have that $\overline{\mathcal{C}}$ is the convex hull of the set of its extreme points, and because it is closed, all its extreme points are also exposed points (Barvinok, 2002). Furthermore, by Caratheodory's theorem, we have that every point in $\overline{\mathcal{C}}$ can be expressed as convex combination of $d+1$ exposed points of $\overline{\mathcal{C}}$. As a result, we have that $\mathbf{C}^* \in \overline{\mathcal{C}}$ can be expressed as a convex combination of $d+1$ exposed points $\mathbf{C}^1, \mathbf{C}^2, \ldots, \mathbf{C}^{d+1} \in \overline{\mathcal{C}}$:

$$\mathbf{C}^* = \sum_{i=1}^{d+1} \lambda_i \mathbf{C}^i,$$

where $\lambda_1, \lambda_2, \ldots \lambda_{d+1} \in \mathbb{R}_+$ are coefficients that sum to 1.

Recall that all exposed points of a convex set are associated with (at least) one hyperplane such that its intersection with the convex set is a singleton set containing only the exposed point. Let $\mathbf{L}_1^*, \mathbf{L}_2^*, \ldots, \mathbf{L}_{d+1}^*$ denote the corresponding hyperplane normals associated with the $d+1$ exposed points $\mathbf{C}^1, \mathbf{C}^2, \ldots, \mathbf{C}^{d+1}$ used to express $\mathbf{C}^*$. Consequently, each function $\langle \mathbf{L}_i^*, \mathbf{C} \rangle$ achieves its unique minimum over $\mathbf{C} \in \overline{\mathcal{C}}$ at $\mathbf{C} = \mathbf{C}^i$.

We also have from Proposition 5 that for each hyperplane $\mathbf{L}_i^*$, there exists a deterministic classifier $h_i : \mathcal{X} \to [n]$ that minimizes the linear performance metric $\langle \mathbf{L}_i^*, \mathbf{C}[h] \rangle$. All that remains to show is that $\mathbf{C}[h_i] = \mathbf{C}^i$. We can then conclude that the randomized classifier $h^* = \sum_{i=1}^{d+1} \lambda_i h_i$ is an optimal classifier for OP2 as $\mathbf{C}[h^*] = \sum_{i=1}^{d+1} \lambda_i \mathbf{C}[h_i] = \sum_{i=1}^{d+1} \lambda_i \mathbf{C}^i = \mathbf{C}^*$ and $\mathbf{C}^* \in \mathcal{C}$.

To prove $\mathbf{C}[h_i] = \mathbf{C}^i$, let us assume the contrary that $\mathbf{C}[h_i] \neq \mathbf{C}^i$. Since $\mathbf{C}^i$ is the unique minimizer of $\langle \mathbf{L}_i^*, \mathbf{C} \rangle$, this would mean that $\langle \mathbf{L}_i^*, \mathbf{C}[h_i] \rangle > \langle \mathbf{L}_i^*, \mathbf{C}^i \rangle$. Suppose we construct a line joining $\mathbf{C}[h_i]$ and $\mathbf{C}^i$; all points on this line lie in $\mathcal{C}$ except for potentially the end point $\mathbf{C}^i$. All the points $\mathbf{C}'$ in the interior of this line must then satisfy: $\langle \mathbf{L}_i^*, \mathbf{C}[h_i] \rangle > \langle \mathbf{L}_i^*, \mathbf{C}' \rangle > \langle \mathbf{L}_i^*, \mathbf{C}^i \rangle$. However, this would say that the classifier corresponding to an interior point $\mathbf{C}'$ has a lower loss than $\mathbf{C}[h_i]$, contradicting the fact that h_i minimizes $\langle \mathbf{L}_i^*, \mathbf{C}[h] \rangle$. By contradiction, we have that $\mathbf{C}[h_i] = \mathbf{C}^i$. □

A.4 Proof of Theorem 12 (Frank-Wolfe for Unconstrained Problems)

Theorem ((Restated) Convergence of FW algorithm)**.** *Fix $\epsilon \in (0,1)$. Let $\psi : [0,1]^d \to [0,1]$ be convex, and β-smooth and L-Lipschitz w.r.t. the ℓ_2-norm. Let Ω in Algorithm 1 be a (ρ, ρ', δ)-approximate LMO for sample size m. Let $\bar{h}$ be a classifier returned by Algorithm 1 when run for T iterations. Then with probability $\geq 1 - \delta$ over draw of $S \sim D^N$, after $T = \mathcal{O}(1/\epsilon)$ iterations:*

$$\psi(\mathbf{C}[\bar{h}]) \leq \min_{\mathbf{C} \in \mathcal{C}} \psi(\mathbf{C}) + \mathcal{O}\left(\beta\epsilon + L\rho + \beta\sqrt{d}\rho'\right).$$

We first prove an important lemma where we bound the approximation error of the linear minimization oracle used in the algorithm. This result coupled with the standard convergence analysis for the Frank-Wolfe method (Jaggi, 2013) will then allow us to prove the above theorem.

Lemma 27. *Let $\psi : [0,1]^d \to \mathbb{R}_+$ be convex over $\mathcal{C}$, and L-Lipschitz and β-smooth w.r.t. the ℓ_2 norm. Let classifiers $\widetilde{h}^1, \ldots, \widetilde{h}^T$, and $h^0, h^1, \ldots, h^T$ be as defined in Algorithm 1. Then for any $\delta \in (0,1]$, with probability at least $1-\delta$ (over draw of S from D^N), we have for all $1 \leq t \leq T$*

$$\langle \nabla\psi(\mathbf{C}[h^{t-1}]), \mathbf{C}[\widetilde{h}^t] \rangle \leq \min_{\mathbf{g}:\mathcal{X}\to\Delta_n} \langle \nabla\psi(\mathbf{C}[h^{t-1}]), \mathbf{C}[\mathbf{g}] \rangle + L\rho + 2\beta\sqrt{d}\rho'.$$

Proof. For any $1 \leq t \leq T$, let $\mathbf{g}^{t,*} \in \operatorname{argmin}_{\mathbf{g}:\mathcal{X}\to\Delta_n} \langle \nabla\psi(\mathbf{C}[h^{t-1}]), \mathbf{C}[\mathbf{g}] \rangle$. We then have

$$\begin{aligned}
&\langle \nabla\psi(\mathbf{C}[h^{t-1}]), \mathbf{C}[\widetilde{h}^t] \rangle - \min_{\mathbf{g}:\mathcal{X}\to\Delta_n} \langle \nabla\psi(\mathbf{C}[h^{t-1}]), \mathbf{C}[\mathbf{g}] \rangle \\
&= \langle \nabla\psi(\mathbf{C}[h^{t-1}]), \mathbf{C}[\widetilde{h}^t] \rangle - \langle \nabla\psi(\mathbf{C}[h^{t-1}]), \mathbf{C}[\mathbf{g}^{t,*}] \rangle \\
&= \underbrace{\langle \nabla\psi(\mathbf{C}[h^{t-1}]), \mathbf{C}[\widetilde{h}^t] - \langle \nabla\psi(\mathbf{C}^{t-1}), \mathbf{C}[\widetilde{h}^t] \rangle}_{\text{term}_1} \\
&\qquad + \underbrace{\langle \nabla\psi(\mathbf{C}^{t-1}), \mathbf{C}[\widetilde{h}^t] \rangle - \langle \nabla\psi(\mathbf{C}^{t-1}), \mathbf{C}[\mathbf{g}^{t,*}] \rangle}_{\text{term}_2} \\
&\qquad + \underbrace{\langle \nabla\psi(\mathbf{C}^{t-1}), \mathbf{C}[\mathbf{g}^{t,*}] \rangle - \langle \nabla\psi(\mathbf{C}[h^{t-1}]), \mathbf{C}[\mathbf{g}^{t,*}] \rangle}_{\text{term}_3}.
\end{aligned}$$

We next bound each of these terms. We start with term$_2$. For any $1 \leq t \leq T$, let $\mathbf{L}^t$ be as defined in Algorithm 1. For all $1 \leq t \leq T$,

$$\begin{aligned}
\langle \nabla\psi(\mathbf{C}^{t-1}), \mathbf{C}[\widetilde{h}^t] \rangle - \langle \nabla\psi(\mathbf{C}^{t-1}), \mathbf{C}[\mathbf{g}^{t,*}] \rangle &= \|\nabla\psi(\mathbf{C}^{t-1})\|_\infty (\langle \mathbf{L}^t, \mathbf{C}[\mathbf{g}^{t,*}] \rangle - \langle \mathbf{L}^t, \mathbf{C}[\widetilde{h}^t] \rangle) \\
&\leq \|\nabla\psi(\mathbf{C}^{t-1})\|_2 (\langle \mathbf{L}^t, \mathbf{C}[\mathbf{g}^{t,*}] \rangle - \langle \mathbf{L}^t, \mathbf{C}[\widetilde{h}^t] \rangle) \\
&\leq L\rho,
\end{aligned}$$

which follows from the property of the LMO (in Definition 11) and from L-Lipchitzness of ψ, and holds with probability at least $1-\delta$ (over draw of S).

Next, for term$_3$, we have by an application of Holder's inequality

$$\begin{aligned}
&\langle \nabla\psi(\mathbf{C}^{t-1}), \mathbf{C}[\mathbf{g}^{t,*}] \rangle - \langle \nabla\psi(\mathbf{C}[h^{t-1}]), \mathbf{C}[\mathbf{g}^{t,*}] \rangle \\
&\leq \left\| \nabla\psi(\mathbf{C}^{t-1}) - \nabla\psi(\mathbf{C}[h^{t-1}]) \right\|_\infty \|\mathbf{C}[\mathbf{g}^{t,*}]\|_1 \\
&= \left\| \nabla\psi(\mathbf{C}^{t-1}) - \nabla\psi(\mathbf{C}[h^{t-1}]) \right\|_\infty (1) \\
&\leq \left\| \nabla\psi(\mathbf{C}^{t-1}) - \nabla\psi(\mathbf{C}[h^{t-1}]) \right\|_2 \\
&\leq \beta \left\| \mathbf{C}^{t-1} - \mathbf{C}[h^{t-1}] \right\|_2 \\
&\leq \beta\sqrt{d} \left\| \mathbf{C}^{t-1} - \mathbf{C}[h^{t-1}] \right\|_\infty \\
&\leq \beta\sqrt{d} \left\| \left(1 - \frac{2}{t}\right) \widetilde{\mathbf{C}}^{t-2} + \frac{2}{t}\widetilde{\mathbf{C}}^{t-1} - \left(1 - \frac{2}{t}\right) \mathbf{C}[h^{t-2}] + \frac{2}{t}\mathbf{C}[h^{t-1}] \right\|_\infty \\
&\leq \beta\sqrt{d} \left(\left(1 - \frac{2}{t}\right) \|\widetilde{\mathbf{C}}^{t-2} - \mathbf{C}[h^{t-2}]\|_\infty + \frac{2}{t} \|\widetilde{\mathbf{C}}^{t-1} - \mathbf{C}[h^{t-1}]\|_\infty \right) \\
&\leq \beta\sqrt{d} \max \left\{ \|\widetilde{\mathbf{C}}^{t-2} - \mathbf{C}[h^{t-2}]\|_\infty, \|\widetilde{\mathbf{C}}^{t-1} - \mathbf{C}[h^{t-1}]\|_\infty \right\} \\
&\quad \vdots \\
&\leq \beta\sqrt{d} \max_{i\in[t]} \left\{ \|\widetilde{\mathbf{C}}^{i-1} - \mathbf{C}[h^{i-1}]\|_\infty \right\} \\
&\leq \beta\sqrt{d}\rho',
\end{aligned}$$

where the fourth step follows from β-smoothness of ψ; the last step uses the property of the LMO and holds with probability at least $1-\delta$ (over draw of S). One can similarly bound term_1.

We thus have for all $1 \leq t \leq T$, with probability at least $1-\delta$ (over draw of S),

$$\langle \nabla\psi(\mathbf{C}[h^{t-1}]), \mathbf{C}[\widetilde{h}^t]\rangle - \min_{\mathbf{g}:\mathcal{X}\to\Delta_n} \langle \nabla\psi(\mathbf{C}[h^{t-1}]), \mathbf{C}[\mathbf{g}]\rangle \quad \leq \quad L\rho + 2\beta\sqrt{d}\rho',$$

as desired. □

We are now ready to prove Theorem 12.

Proof of Theorem 12. Our proof shall make use of the standard convergence result for the Frank-Wolfe algorithm for minimizing a convex function over a convex set (Jaggi, 2013). We will find it useful to first define the following quantity, referred to as the curvature constant in Jaggi (2013).

$$C_\psi \quad = \quad \sup_{\mathbf{C}_1,\mathbf{C}_2\in\mathcal{C},\gamma\in[0,1]} \frac{2}{\gamma^2}\Big(\psi\big(\mathbf{C}_1+\gamma(\mathbf{C}_2-\mathbf{C}_1)\big) - \psi\big(\mathbf{C}_1\big) - \gamma\big\langle \mathbf{C}_2-\mathbf{C}_1, \nabla\psi(\mathbf{C}_1)\big\rangle\Big).$$

Also, define two positive scalars ϵ_S and δ_{apx} required in the analysis of Jaggi (2013):

$$\begin{aligned}\epsilon_S &= L\rho + 2\beta\sqrt{d}\rho' \\ \delta_{\text{apx}} &= \frac{(T+1)\epsilon_S}{C_\psi},\end{aligned}$$

where $\delta \in (0,1]$ is as in the theorem statement. Further, let the classifiers $\widetilde{h}^1,\ldots,\widetilde{h}^T$, and $h^0,\ldots,h^T$ be as defined in Algorithm 1. We then have from Lemma 27 that the following holds with probability at least $1-\delta$, for all $1 \leq t \leq T$,

$$\begin{aligned}\left\langle \nabla\psi\left(\mathbf{C}[h^{t-1}]\right), \mathbf{C}[\widetilde{h}^t]\right\rangle &\leq \min_{\mathbf{g}:\mathcal{X}:\to\Delta_n} \left\langle \nabla\psi\left(\mathbf{C}\left[h^{t-1}\right]\right), \mathbf{C}\left[\mathbf{g}\right]\right\rangle + \epsilon_S \\ &= \min_{\mathbf{C}\in\mathcal{C}} \left\langle \nabla\psi\left(\mathbf{C}\left[h^{t-1}\right]\right), \mathbf{C}\right\rangle + \epsilon_S \\ &= \min_{\mathbf{C}\in\mathcal{C}} \left\langle \nabla\psi\left(\mathbf{C}\left[h^{t-1}\right]\right), \mathbf{C}\right\rangle + \frac{1}{2}\delta_{\text{apx}}\frac{2}{T+1}C_\psi \\ &\leq \min_{\mathbf{C}\in\mathcal{C}} \langle \nabla\psi\left(\mathbf{C}\left[h^{t-1}\right]\right), \mathbf{C}\rangle + \frac{1}{2}\delta_{\text{apx}}\frac{2}{t+1}C_\psi\,. \qquad (11)\end{aligned}$$

Also observe that for the two sequences of iterates given by the confusion matrices of the above classifiers,

$$\mathbf{C}[h^t] \quad = \quad \left(1-\frac{2}{t+1}\right)\mathbf{C}[h^{t-1}] + \frac{2}{t+1}\mathbf{C}[\widetilde{h}^t], \qquad (12)$$

for all $1 \leq t \leq T$. Based on (11) and (12), one can now apply the result of Jaggi (2013).

In particular, the sequence of iterates $\mathbf{C}[h^0], \mathbf{C}[h^1], \ldots, \mathbf{C}[h^T]$ can be considered as the sequence of iterates arising from running the Frank-Wolfe optimization method to minimize ψ over $\overline{\mathcal{C}}$ with a linear optimization oracle that is $\frac{1}{2}\delta_{\text{apx}}\frac{2}{t+1}C_\psi$ accurate at iteration t. Since ψ is a convex

function over the convex constraint set $\mathcal{C}$, one has from Theorem 1 in Jaggi (2013) that the following convergence guarantee holds with probability at least $1-\delta$:

$$\begin{aligned}\psi(\mathbf{C}[\bar{h}]) \;=\; \psi(\mathbf{C}[h^T]) \;&\leq\; \min_{\mathbf{C}\in\mathcal{C}} \psi(\mathbf{C}) + \frac{2C_\psi}{T+2}(1+\delta_{\text{apx}}) \\ &=\; \min_{\mathbf{C}\in\mathcal{C}} \psi(\mathbf{C}) + \frac{2C_\psi}{T+2} + \frac{2\epsilon_S(T+1)}{T+2} \\ &\leq\; \min_{\mathbf{C}\in\mathcal{C}} \psi(\mathbf{C}) + \frac{2C_\psi}{T+2} + 2\epsilon_S \qquad (13)\end{aligned}$$

We can further upper bound C_ψ in terms of the the smoothness parameter of ψ:

$$\begin{aligned}C_\psi \;&=\; \sup_{\mathbf{C}_1,\mathbf{C}_2\in\mathcal{C},\gamma\in[0,1]} \frac{2}{\gamma^2}\Big(\psi\big(\mathbf{C}_1+\gamma(\mathbf{C}_2-\mathbf{C}_1)\big) - \psi\big(\mathbf{C}_1\big) - \gamma\big\langle \mathbf{C}_2-\mathbf{C}_1, \nabla\psi(\mathbf{C}_1)\big\rangle\Big) \\ &\leq\; \sup_{\mathbf{C}_1,\mathbf{C}_2\in\mathcal{C},\gamma\in[0,1]} \frac{2}{\gamma^2}\Big(\frac{\beta}{2}\gamma^2||\mathbf{C}_1-\mathbf{C}_2||_2^2\Big) \;=\; 4\beta\;,\end{aligned}$$

where the second step follows from the β-smoothness of ψ. Substituting back in (13), we finally have with probability at least $1-\delta$,

$$\psi(\mathbf{C}[\bar{h}]) \;\leq\; \min_{\mathbf{C}\in\mathcal{C}} \psi(\mathbf{C}) + \frac{8\beta}{T+2} + 2\epsilon_S \;=\; \min_{\mathbf{C}\in\mathcal{C}} \psi(\mathbf{C}) + \frac{8\beta}{T+2} + 2L\rho + 4\beta\sqrt{d}\rho'.$$

Setting $T = 1/\epsilon$ completes the proof. □

A.5 Proof of Theorem 13 (GDA for Unconstrained Problems)

Theorem 13 follows from Theorem 17 under the special case of $K = 0$. The algorithms for the constrained and unconstrained case become identical and the same bounds apply with $r = \infty$. Please see Appendix A.9 for proof of Theorem 17.

A.6 Proof of Theorem 14 (Ellipsoid For Unconstrained Problems)

Theorem 14 follows from Theorem 18 under the special case of $K = 0$. The algorithms for the constrained and unconstrained case become identical and the same bounds apply with $r = \infty$. Please see Appendix A.10 for proof of Theorem 18.

A.7 Proof of Theorem 15 (Bisection For Unconstrained Problems)

Theorem 15 follows from Theorem 19 under the special case of $K = 0$ and $T' = 1$. Please see Appendix A.11 for proof of Theorem 19.

A.8 Proof of Theorem 16 (SplitFW for Constrained Problems)

Theorem ((Restated) Convergence of SplitFW algorithm)**.** *Fix $\epsilon > 0$. Let $\psi : [0,1]^d \to [0,1]$ be convex, β-smooth and L-Lipschitz w.r.t. the ℓ_2-norm, and let $\phi_1, \ldots, \phi_K : [0,1]^d \to [-1,1]$ be convex and L-Lipschitz w.r.t. the ℓ_2-norm. Let Ω in Algorithm 5 be a (ρ, ρ', δ)-approximate LMO for sample size N. Let $\bar{h}$ be a classifier returned by Algorithm 5 when run for T iterations with some $\zeta > 0$. Let the strict feasibility condition in Assumption 1 hold for radius $r > 0$. Then, with probability*

$\geq 1-\delta$ over draw of $S \sim D^N$, after $T = \mathcal{O}(1/\epsilon^2)$ iterations, the classifier $\bar{h}$ is near-optimal and near-feasible:

$$\textbf{\textit{Optimality:}}\ \psi(\mathbf{C}[\bar{h}]) \leq \min_{\mathbf{C}\in\mathcal{C},\phi_k(\mathbf{C})\leq 0,\forall k} \psi(\mathbf{C}) + \mathcal{O}\left(\epsilon + \sqrt{\rho^{\text{eff}}}\right);$$

$$\textbf{\textit{Feasibility:}}\ \phi_k(\mathbf{C}[\bar{h}]) \leq \mathcal{O}\left(\epsilon + \sqrt{\rho^{\text{eff}}}\right),\ \forall k \in [K].$$

where $\rho^{\text{eff}} = \rho + \sqrt{d}\rho'$ and the $\mathcal{O}$ notation hides constant factors independent of $\rho, \rho', T, \epsilon, d, K$ for small enough ρ, ρ' and large enough T (or small enough ϵ).

There are two key steps to the proof of this theorem. First, we show that the use of an approximate LMO in steps 9, 11 of Algorithm 5 does not affect the convergence results by Gidel et al. (2018). Specifically, they measure the sub-optimality of an iterate using a duality gap measure. In Lemma 32 we show that a similar bound on the duality gap can be derived with an approximate LMO over $\mathcal{C}$. Second, we use the strict feasibilty assumption to convert a bound on the duality gap into a bound on the sub-optimality of the in problem (4) in Lemma 31.

We will find it useful to first define the following quantities: fat achievable set, dual functions, and the primal and dual gaps.

Definition 28 (Fat achievable set)**.** *The set $\mathcal{C}_{\rho'}$ is defined as follows:*

$$\mathcal{C}_{\rho'} = \left(\mathcal{C} + B(\mathbf{0}, \sqrt{d}\rho')\right) \cap \Delta_d = \{\mathbf{C} + \mathbf{r} : \mathbf{C} \in \mathcal{C}, \|\mathbf{r}\|_2 \leq \sqrt{d}\rho', \mathbf{C} + \mathbf{r} \in \Delta_d\}.$$

The set $\mathcal{C}_{\rho'}$ is defined so that the iterates $\widetilde{\mathbf{C}}^t$ and $\mathbf{C}^t$ lie within $\mathcal{C}'_\rho$ with high probability.

Definition 29 (Dual function)**.** *The dual function $f^{\text{aug}} : \mathbb{R}^d \to \mathbb{R}$ is defined as*

$$f^{\text{aug}}(\boldsymbol{\lambda}) = \min_{\mathbf{C}\in\mathcal{C}_{\rho'},\mathbf{F}\in\mathcal{F}} \mathcal{L}^{\text{aug}}(\mathbf{C}, \mathbf{F}, \boldsymbol{\lambda}).$$

We also use $\widehat{\mathbf{C}}(\boldsymbol{\lambda}), \widehat{\mathbf{F}}(\boldsymbol{\lambda})$ to denote any arbitrary minimizer of $\mathcal{L}^{\text{aug}}(.,.,\boldsymbol{\lambda})$ over $\mathcal{C}_{\rho'} \times \mathcal{F}$. Thus $f^{\text{aug}}(\boldsymbol{\lambda}) = \mathcal{L}^{\text{aug}}(\widehat{\mathbf{C}}(\boldsymbol{\lambda}), \widehat{\mathbf{F}}(\boldsymbol{\lambda}), \boldsymbol{\lambda})$. Further, let the maximum value of the dual function be $f^{\text{aug}*}$. By the min-max theorem, we have that

$$f^{\text{aug}*} = \max_{\boldsymbol{\lambda}\in\mathbb{R}^d} \min_{\mathbf{C}\in\mathcal{C}_{\rho'},\mathbf{F}\in\mathcal{F}} \mathcal{L}^{\text{aug}}(\mathbf{C}, \mathbf{F}, \boldsymbol{\lambda}) = \min_{\mathbf{C}\in\mathcal{C}_{\rho'},\mathbf{F}\in\mathcal{F}} \max_{\boldsymbol{\lambda}\in\mathbb{R}^d} \mathcal{L}^{\text{aug}}(\mathbf{C}, \mathbf{F}, \boldsymbol{\lambda}) = \min_{\mathbf{C}\in\mathcal{C}_{\rho'}\cap\mathcal{F}} 2\psi(\mathbf{C}).$$

The last equality follows from the observation that if $\mathbf{C} \neq \mathbf{F}$ then $\max_{\boldsymbol{\lambda}\in\mathbb{R}^d} \mathcal{L}^{\text{aug}}(\mathbf{C}, \mathbf{F}, \boldsymbol{\lambda}) = \infty$. Next, let $\mathbf{C}^* \in \mathcal{C}_{\rho'} \cap \mathcal{F}$ such that

$$\psi(\mathbf{C}^*) = \min_{\mathbf{C}\in\mathcal{C}_{\rho'}\cap\mathcal{F}} \psi(\mathbf{C}).$$

and let $\mathcal{W}^* = \text{argmax}_{\boldsymbol{\lambda}\in\mathbb{R}^d} f^{\text{aug}}(\boldsymbol{\lambda}) \subseteq \mathbb{R}^d$.

Definition 30 (Primal and dual gaps)**.** *For any $\mathbf{C} \in \mathcal{C}_{\rho'}, \mathbf{F} \in \mathcal{F}$ and $\boldsymbol{\lambda} \in \mathbb{R}^d$, we define the primal and dual gaps as follows:*

$$\Delta^{(p)}(\mathbf{C}, \mathbf{F}, \boldsymbol{\lambda}) = \mathcal{L}^{\text{aug}}(\mathbf{C}, \mathbf{F}, \boldsymbol{\lambda}) - \min_{\mathbf{C}\in\mathcal{C}_{\rho'},\mathbf{F}\in\mathcal{F}} \mathcal{L}^{\text{aug}}(\mathbf{C}, \mathbf{F}, \boldsymbol{\lambda}) = \mathcal{L}^{\text{aug}}(\mathbf{C}, \mathbf{F}, \boldsymbol{\lambda}) - f^{\text{aug}}(\boldsymbol{\lambda});$$

$$\Delta^{(d)}(\boldsymbol{\lambda}) = f^{\text{aug}*} - f^{\text{aug}}(\boldsymbol{\lambda}) = 2\psi(\mathbf{C}^*) - f^{\text{aug}}(\boldsymbol{\lambda}),$$

and define the total gap as $\Delta(\mathbf{C}, \mathbf{F}, \boldsymbol{\lambda}) = \Delta^{(p)}(\mathbf{C}, \mathbf{F}, \boldsymbol{\lambda}) + \Delta^{(d)}(\boldsymbol{\lambda})$.

In the theorems and lemmas below, we will refer to the iterates $\mathbf{C}^t, \mathbf{F}^t, \widetilde{\mathbf{C}}^t, \widetilde{\mathbf{F}}^t$ in the Algorithm 5. We use the the short-hands $\Delta_t, \Delta_t^{(\mathrm{p})}, \Delta_t^{(\mathrm{d})}$ for representing the same primal and dual gaps evaluated at, $(\mathbf{C}^{t+1}, \mathbf{F}^{t+1}, \boldsymbol{\lambda}^t)$.

We will require the use of Theorem 1 and Corollary 1 from Gidel et al. (2018), which we restate below in our notation. We use the following facts to transform their Theorem. The norms of vectors correspond to the ℓ_2-norm unless specified otherwise. We also overload notation and refer to the concatenation of two vectors $\mathbf{C}, \mathbf{F}$ as $[\mathbf{C}, \mathbf{F}]$.

$$
\begin{aligned}
|\psi(\mathbf{C}) + \psi(\mathbf{F}) - \psi(\mathbf{C}') - \psi(\mathbf{F}')| &\leq 2L\|[\mathbf{C} - \mathbf{C}', \mathbf{F} - \mathbf{F}']\|_2 \\
\max\left(\text{eigen-val}\left([I, -I]^\top [-I, I]\right)\right) &= 2 \\
(\text{diam}(\mathcal{F}))^2 &\leq \text{diam}(\Delta_d)^2 \leq 2 \\
(\text{diam}(\mathcal{C}_{\rho'}))^2 &\leq \text{diam}(\Delta_d)^2 \leq 2 \\
(\text{diam}(\mathcal{C}_{\rho'} \times \mathcal{F}))^2 &\leq 4,
\end{aligned}
$$

where $\|M\|$ of a matrix M refers to its spectral norm, and $\text{diam}(\mathcal{A})$ refers to the diameter of a set $\mathcal{A}$, i.e. the maximum ℓ_2 distance between any two elements from the set $\mathcal{A}$.

Theorem (Restated from Gidel et al. (2018))**.** *There exists a constant $\alpha > 0$ such that*

$$
\begin{aligned}
f^{\text{aug}*} - f^{\text{aug}}(\boldsymbol{\lambda}) &\geq \frac{1}{8L_\zeta} \min\left\{\alpha^2 \text{dist}(\boldsymbol{\lambda}, \mathcal{W}^*)^2, \alpha L_\zeta Z^2 \text{dist}(\boldsymbol{\lambda}, \mathcal{W}^*)\right\}; \\
||\nabla f^{\text{aug}}(\boldsymbol{\lambda})||_2 &\geq \frac{1}{8L_\zeta} \min\left\{\alpha^2 \text{dist}(\boldsymbol{\lambda}, \mathcal{W}^*), \alpha L_\zeta Z^2\right\}; \\
||\nabla f^{\text{aug}}(\boldsymbol{\lambda})||_2 &\geq \frac{\alpha}{\sqrt{8L_\zeta}} \min\left\{\sqrt{f^{\text{aug}*} - f^{\text{aug}}(\boldsymbol{\lambda})}, \sqrt{\frac{L_\zeta Z^2}{2}}\right\},
\end{aligned}
$$

where $L_\zeta = 2L + 2\zeta$ and dist *represents the standard distance function between a point and a set, i.e. dist$(\mathbf{x}, \mathcal{A}) = \min_{\mathbf{x}' \in \mathcal{A}} \|\mathbf{x} - \mathbf{x}'\|$.*

We will fix a probability of failure δ throughout the rest of the proof, and assume that the training sample S is "good", in which case the empirical confusion matrix output by the Ω is ρ' close to the true confusion matrix of the classifier whenever it is called by Algorithm 5.

We then show below that, if the total gap is low then the resulting classifier is close to optimal and feasible.

Lemma 31. *Let the assumptions in Theorem 16 hold. Let $g : \mathcal{X} \to \Delta_n$ be a randomized classifier, and $\mathbf{C} \in \Delta_d$ be such that $\|\mathbf{C} - \mathbf{C}[g]\|_\infty \leq \rho'$. Let $\mathbf{F} \in \mathcal{F}, \boldsymbol{\lambda} \in \mathbb{R}^d$ be such that $\Delta(\mathbf{C}, \mathbf{F}, \boldsymbol{\lambda}) \leq \tau$ and $||\mathbf{C} - \mathbf{F}||_2^2 \leq \kappa$. We then have:*

$$
\begin{aligned}
\psi(C[g]) &\leq \min_{\mathbf{C}' \in \mathcal{C} \cap \mathcal{F}} \psi(\mathbf{C}') + \frac{\tau}{2} + (\gamma + L)\sqrt{\kappa} + L\sqrt{d}\rho' \\
\|\boldsymbol{\phi}(\mathbf{C}[g])\|_\infty &\leq L(\sqrt{d}\rho' + \sqrt{\kappa}),
\end{aligned}
$$

where $\gamma = \frac{2L}{r} + \frac{\zeta r}{2L} + \frac{\tau L}{r}$.

Proof. The second inequality in the lemma trivially follows from the triangle inequality and the ℓ_2 Lipschitzness of the constraint functions ϕ_k, i.e. for any $k \in [K]$

$$\begin{aligned}\phi_k(\mathbf{C}[g]) &\leq \phi_k(\mathbf{C}) + L\|\mathbf{C} - \mathbf{C}[g]\|_2 \\ &\leq \phi_k(\mathbf{C}) + L\sqrt{d}\rho' \\ &\leq \phi_k(\mathbf{F}) + L\|\mathbf{F} - \mathbf{C}\|_2 + L\sqrt{d}\rho' \\ &\leq L\sqrt{\kappa} + L\sqrt{d}\rho'\end{aligned}$$

We will prove the first inequality below. By construction, $\mathbf{C} \in \mathcal{C}_{\rho'}$. As $\Delta(\mathbf{C}, \mathbf{F}, \boldsymbol{\lambda}) \leq \tau$, we have

$$\Delta^{(\mathrm{p})}(\mathbf{C}, \mathbf{F}, \boldsymbol{\lambda}) = \mathcal{L}^{\mathrm{aug}}(\mathbf{C}, \mathbf{F}, \boldsymbol{\lambda}) - \min_{\mathbf{C}' \in \mathcal{C}_{\rho'}, \mathbf{F}' \in \mathcal{F}} \mathcal{L}^{\mathrm{aug}}(\mathbf{C}', \mathbf{F}', \boldsymbol{\lambda}) \leq \tau \tag{14}$$

$$\Delta^{(\mathrm{d})}(\boldsymbol{\lambda}) = 2\psi(\mathbf{C}^*) - \min_{\mathbf{C}' \in \mathcal{C}_{\rho'}, \mathbf{F}' \in \mathcal{F}} \mathcal{L}^{\mathrm{aug}}(\mathbf{C}', \mathbf{F}', \boldsymbol{\lambda}) \leq \tau \tag{15}$$

where $\mathbf{C}^* \in \operatorname{argmin}_{\mathbf{C}' \in \mathcal{C}_{\rho'} \cap \mathcal{F}} \psi(\mathbf{C}')$. Setting $\mathbf{C}' = \mathbf{F}' = \mathbf{C}^*$ in the second term of Eqn. (14):

$$\psi(\mathbf{C}) + \psi(\mathbf{F}) + \boldsymbol{\lambda}^T(\mathbf{C} - \mathbf{F}) + \frac{\zeta}{2}\|\mathbf{C} - \mathbf{F}\|_2^2 \leq 2\psi(\mathbf{C}^*) + \tau\,. \tag{16}$$

The variables $\mathbf{C}', \mathbf{F}'$ in the second term of (15) are set as follows. Let $\mathbf{C}' = \mathbf{C}[h]$ be a strictly feasible point, i.e. $\boldsymbol{\phi}(\mathbf{C}') \leq -r$. Such a h exists by Assumption 1. As the constraint functions ϕ_k are all L-Lipschitz w.r.t. ℓ_2 norm, a ball of radius $\frac{r}{L}$ centered at $\mathbf{C}'$ is a subset of $\mathcal{F}$. Further, let $\mathbf{F}' = \mathbf{C}' + \frac{r}{L||\boldsymbol{\lambda}||}\boldsymbol{\lambda}$. We then have:

$$2\psi(\mathbf{C}^*) \leq \psi(\mathbf{C}') + \psi(\mathbf{F}') - \frac{r\|\boldsymbol{\lambda}\|_2}{L} + \frac{\zeta r^2}{2L^2} + \tau. \tag{17}$$

This can be reduced to a bound on $||\boldsymbol{\lambda}||_2$,

$$\|\boldsymbol{\lambda}\|_2 \leq \frac{2L}{r} + \frac{\zeta r}{2L} + \frac{\tau L}{r} = \gamma. \tag{18}$$

From Cauchy-Schwarz inequality, (16) becomes:

$$\begin{aligned}\psi(\mathbf{C}) + \psi(\mathbf{F}) &\leq 2\psi(\mathbf{C}^*) + \tau - \boldsymbol{\lambda}^\top(\mathbf{C} - \mathbf{F}) - \frac{\zeta}{2}\|\mathbf{C} - \mathbf{F}\|_2^2 \\ &\leq 2\psi(\mathbf{C}^*) + \tau + \gamma\sqrt{\kappa}.\end{aligned} \tag{19}$$

As ψ is L-Lipschitz, we have

$$\psi(\mathbf{C}) - \psi(\mathbf{F}) \leq L\|\mathbf{C} - \mathbf{F}\|_2 \leq L\sqrt{\kappa}. \tag{20}$$

Adding (19) and (20) and dividing by 2, we get

$$\psi(\mathbf{C}) \leq \min_{\mathbf{C}' \in \mathcal{C}_{\rho'} \cap \mathcal{F}} \psi(\mathbf{C}') + \frac{\tau}{2} + (\gamma + L)\sqrt{\kappa}.$$

As $\mathcal{C}_{\rho'} \supseteq \mathcal{C}$, and ψ is L-Lipschitz, we have

$$\begin{aligned}\psi(C[\mathbf{g}]) &\leq \psi(\mathbf{C}) + L\|\mathbf{C} - C[\mathbf{g}]\|_2 \\ &\leq \min_{\mathbf{C}' \in \mathcal{C}_{\rho'} \cap \mathcal{F}} \psi(\mathbf{C}') + \frac{\tau}{2} + (\gamma + L)\sqrt{\kappa} + L\sqrt{d}\rho' \\ &\leq \min_{\mathbf{C}' \in \mathcal{C} \cap \mathcal{F}} \psi(\mathbf{C}') + \frac{\tau}{2} + (\gamma + L)\sqrt{\kappa} + L\sqrt{d}\rho',\end{aligned}$$

which completes the proof. □

The lemma below bounds the duality gap Δ_t and $\|\mathbf{C}_t - \mathbf{F}_t\|^2$ based on the proof of Theorem 2 in Gidel et al. (2018). The only difference is the approximate nature of the LMO, that simply contributes an additive factor of $\mathcal{O}(\rho + \sqrt{d}\rho')$ to the convergence rate of $\mathcal{O}(1/t)$. The proof is highly technical, and we skip it for brevity. The details can be inferred from Tavker et al. (2020), which contains the full proof using a different notation.

Lemma 32. *Let the assumptions in Theorem 16 hold. Let $t_* \in [T]$ be such that $\bar{h} = h^{t_*}$ in Algorithm 5. Let Ω be a (ρ, ρ', δ)-approximate LMO. For large enough T and ζ, with probability $1 - \delta$ over draw of $S \sim D^N$ we have that*

$$\begin{aligned}\Delta(\mathbf{C}_{t_*}, \mathbf{F}_{t_*}, \boldsymbol{\lambda}_{t_*-1}) &\leq c_1(\rho + \sqrt{d}\rho') + \frac{c_2}{T}; \\ ||\mathbf{C}_{t_*} - \mathbf{F}_{t_*}||_2^2 &\leq c_3(\rho + \sqrt{d}\rho') + \frac{c_4}{T},\end{aligned}$$

where $h_{t_*}, \mathbf{F}_{t_*}, \boldsymbol{\lambda}_{t_*-1}$ are as defined in Algorithm 5. The constants c_1, c_2, c_3 and c_4 are independent of the dimension d and number of constraints K, approximation constants ρ, ρ' and iterations T. More explicitly, $c_1 = \frac{4+12\zeta}{a\zeta}$, $c_2 = 16(\beta + 2\zeta)(t_0 + 2)$, $c_3 = \frac{8+24\zeta}{\zeta}\left[1 + \frac{2}{a}\right]$, $c_4 = 8\left[32(\beta + 2\zeta)\frac{(t_0+2)}{a} + \frac{64a(\beta+2\zeta)}{\zeta^2}\right]$, $a = \min\left[\frac{2}{\zeta}, \frac{\alpha^2}{8(\beta+2\zeta)}\right]$, and t_0 is a constant > 0.

We are now ready to prove Theorem 16.

Proof of Theorem 16. We first note that Lemma 32 can be applied to Lemma 31 setting $\tau = c_1\left(\rho + \sqrt{d}\rho'\right) + \frac{c_2}{T}$ and $\kappa = c_3\left(\rho + \sqrt{d}\rho'\right) + \frac{c_4}{T}$, with the classifier g in Lemma 31 set to the classifer $\bar{h}$ returned by Algorithm 5. For the sake of simplicity, the bound below focuses on the small ρ, ρ' and large T regime. For small enough ρ, ρ' and large enough T, we have $(\gamma + L)\sqrt{\kappa} > \tau + L\sqrt{d}\rho'$, based on the simple argument that for a small enough positive scalar u , we have $c\sqrt{u} > u$. Thus, from the first inequality of Lemma 31,

$$\begin{aligned}\psi(\mathbf{C}[\bar{h}]) &\leq \min_{\mathbf{C}' \in \mathcal{C} \cap \mathcal{F}} \psi(\mathbf{C}') + \frac{\tau}{2} + L\sqrt{d}\rho' + (\gamma + L)\sqrt{\kappa} \\ &\leq \min_{\mathbf{C}' \in \mathcal{C} \cap \mathcal{F}} \psi(\mathbf{C}') + 2(\gamma + L)\sqrt{\kappa} \\ &\leq \min_{\mathbf{C} \in \mathcal{C}, \phi_k(\mathbf{C}) \leq 0, \forall k} \psi(\mathbf{C}) + 2(\gamma + L)\left(\sqrt{c_3\left(\rho + \sqrt{d}\rho'\right) + \frac{c_4}{T}}\right) \\ &\leq \min_{\mathbf{C} \in \mathcal{C}, \phi_k(\mathbf{C}) \leq 0, \forall k} \psi(\mathbf{C}) + 2(\gamma + L)\left(\sqrt{c_3\left(\rho + \sqrt{d}\rho'\right)} + \sqrt{\frac{c_4}{T}}\right)\end{aligned}$$

$$\leq \min_{\mathbf{C}\in\mathcal{C},\phi_k(\mathbf{C})\leq 0,\forall k} \psi(\mathbf{C}) + \mathcal{O}\left(\epsilon + \sqrt{\rho + \sqrt{d}\rho'}\right)$$

By a similar analysis as above, from the second inequality of Lemma 31, we have for small enough ρ, ρ' and large enough T,

$$\begin{aligned}
\phi_k(\mathbf{C}[\bar{h}]) &\leq L\sqrt{d}\rho' + L\sqrt{\kappa} \\
&\leq 2L\sqrt{\kappa} \\
&\leq 2L\left(\sqrt{c_3\left(\rho + \sqrt{d}\rho'\right) + \frac{c_4}{T}}\right) \\
&\leq \mathcal{O}\big(\epsilon + \sqrt{\rho + \sqrt{d}\rho'}\big),
\end{aligned}$$

as desired.

□

A.9 Proof of Theorem 17 (GDA for Constrained Problems)

Theorem ((Restated) Convergence of ConGDA algorithm). *Fix $\epsilon \in (0,1)$. Let $\psi : [0,1]^d \to [0,1]$ and $\phi_1, \ldots, \phi_K : [0,1]^d \to [-1,1]$ be convex and L-Lipschitz w.r.t. the ℓ_2-norm. Let Ω in Algorithm 6 be a (ρ, ρ', δ)-approximate LMO for sample size N. Suppose the strict feasibility condition in Assumption 1 holds for radius $r > 0$. Let the space of Lagrange multipliers $\Lambda = \{\boldsymbol{\lambda} \in \mathbb{R}^d \,|\, \|\boldsymbol{\lambda}\|_2 \leq 2L(1+1/r)\}$, and $\Xi = \{\boldsymbol{\mu} \in \mathbb{R}^K_+ \,|\, \|\boldsymbol{\mu}\|_1 \leq 2/r\}$. Let $B_{\boldsymbol{\phi}} \geq \max_{\boldsymbol{\xi}\in\Delta_d} \|\boldsymbol{\phi}(\boldsymbol{\xi})\|_2$. Let $\bar{h}$ be a classifier returned by Algorithm 6 when run for T iterations, with step-sizes $\eta = \frac{1}{\bar{L}\sqrt{2T}}$ and $\eta' = \frac{\bar{L}}{(1+2\sqrt{K})\sqrt{2T}}$, where $\bar{L} = 4(1+1/r)L + 2/r$. Then with probability $\geq 1-\delta$ over draw of $S \sim D^N$, after $T = \mathcal{O}(K/\epsilon^2)$ iterations:*

$$\psi(\mathbf{C}[\bar{h}]) \;\leq\; \min_{\mathbf{C}\in\mathcal{C}:\,\boldsymbol{\phi}(\mathbf{C})\leq\mathbf{0}} \psi(\mathbf{C}) \;+\; \mathcal{O}\left(L(\epsilon + \rho^{\text{eff}})\right)$$

$$\phi_k(\mathbf{C}[\bar{h}]) \;\leq\; \mathcal{O}\left(L(\epsilon + \rho^{\text{eff}})\right), \forall k \in [K],$$

where $\rho^{\text{eff}} = \rho + \sqrt{d}\rho'$ and the $\mathcal{O}$ notation hides constant factors independent of ρ, ρ', T, d and K.

The proof is an adaptation of the proof of convergence in Narasimhan et al. (2019) for their oracle-based optimizer (Theorem 3 in their paper), but takes into account three differences: (i) they consider a generic objective function that is independent of $\mathbf{C}$, (ii) they assume that ϕ_ks are monotonic, (iii) they perform a full optimization on $\boldsymbol{\xi}$ instead of gradient-based updates. Moreover, unlike them, we employ exponentiated gradient updates, and derive a better dependence on dimension.

We will first find it useful to first state the following lemma, which adapts the proof steps from Lemmas 2, 4 and 6 in Narasimhan et al. (2019).

Lemma 33. *Let $\psi : [0,1]^d \to [0,1]$ and $\phi_1, \ldots, \phi_K : [0,1]^d \to [-1,1]$ be convex and L-Lipschitz w.r.t. the ℓ_2-norm. Suppose the strict feasibility condition in Assumption 1 holds for radius $r > 0$. Let $\mathbf{C}^* \in \operatorname{argmin}_{\mathbf{C}\in\mathcal{C}:\,\boldsymbol{\phi}(\mathbf{C})\leq\mathbf{0}} \psi(\mathbf{C})$, and let*

$$(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in \operatorname*{argmax}_{\boldsymbol{\lambda}\in\mathbb{R}^d, \boldsymbol{\mu}\in\mathbb{R}^K_+} \left\{ \min_{\mathbf{C}\in\mathcal{C},\,\boldsymbol{\xi}\in\Delta^d} \mathcal{L}^{\text{con}}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \right\}.$$

Then:

1. $\psi(\mathbf{C}^*) = \min\limits_{\mathbf{C}\in\mathcal{C},\,\boldsymbol{\xi}\in\Delta_d} \max\limits_{\boldsymbol{\lambda}\in\mathbb{R}^d,\,\boldsymbol{\mu}\in\mathbb{R}^K_+} \mathcal{L}^{\text{con}}(\mathbf{C},\boldsymbol{\xi},\boldsymbol{\lambda},\boldsymbol{\mu}) = \max\limits_{\boldsymbol{\lambda}\in\mathbb{R}^d,\,\boldsymbol{\mu}\in\mathbb{R}^K_+} \min\limits_{\mathbf{C}\in\mathcal{C},\,\boldsymbol{\xi}\in\Delta_d} \mathcal{L}^{\text{con}}(\mathbf{C},\boldsymbol{\xi},\boldsymbol{\lambda},\boldsymbol{\mu})$;
2. $\psi(\mathbf{C}') = \max\limits_{\boldsymbol{\lambda}\in\Lambda} \min\limits_{\boldsymbol{\xi}\in\Delta_d} \mathcal{L}(\mathbf{C}',\boldsymbol{\xi},\boldsymbol{\lambda}) = \min\limits_{\boldsymbol{\xi}\in\Delta_d} \max\limits_{\boldsymbol{\lambda}\in\Lambda} \mathcal{L}(\mathbf{C}',\boldsymbol{\xi},\boldsymbol{\lambda})$, *for any* $\mathbf{C}' \in \mathcal{C}$;
3. $\|\boldsymbol{\mu}^*\|_1 \leq 1/r$;
4. $\|\boldsymbol{\lambda}^*\|_2 \leq L(1+1/r)$.

Proof. For part 1, we begin by writing out the Lagrangian from (2):

$$\mathcal{L}^{\text{con}}(\mathbf{C},\boldsymbol{\xi},\boldsymbol{\lambda},\boldsymbol{\mu}) = \psi(\boldsymbol{\xi}) + \langle\boldsymbol{\lambda}, \mathbf{C} - \boldsymbol{\xi}\rangle + \langle\boldsymbol{\mu}, \boldsymbol{\phi}(\boldsymbol{\xi})\rangle.$$

Since $\mathcal{L}^{\text{con}}$ is convex in $\boldsymbol{\xi}$ and linear in $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, strong duality holds, and we have:

$$\begin{aligned}
\max_{\boldsymbol{\lambda}\in\mathbb{R}^d,\,\boldsymbol{\mu}\in\mathbb{R}^K_+} \min_{\mathbf{C}\in\mathcal{C},\,\boldsymbol{\xi}\in\Delta_d} \mathcal{L}^{\text{con}}(\mathbf{C},\boldsymbol{\xi},\boldsymbol{\lambda},\boldsymbol{\mu}) &= \min_{\mathbf{C}\in\mathcal{C},\,\boldsymbol{\xi}\in\Delta_d} \max_{\boldsymbol{\lambda}\in\mathbb{R}^d,\,\boldsymbol{\mu}\in\mathbb{R}^K_+} \mathcal{L}^{\text{con}}(\mathbf{C},\boldsymbol{\xi},\boldsymbol{\lambda},\boldsymbol{\mu}) \\
&= \min_{\mathbf{C}\in\mathcal{C},\,\boldsymbol{\xi}\in\Delta_d:\,\boldsymbol{\xi}=\mathbf{C},\,\boldsymbol{\phi}(\boldsymbol{\xi})\leq\mathbf{0}} \psi(\mathbf{C}) \\
&= \min_{\mathbf{C}\in\mathcal{C},:\,\boldsymbol{\phi}(\mathbf{C})\leq\mathbf{0}} \psi(\mathbf{C}) \;=\; \psi(\mathbf{C}^*).
\end{aligned}$$

For part 2, we follow similar steps as part 1 except that it applies to the Lagrangian in (2) for the unconstrained problem.

For part 3, recall from our strict feasibility assumption that there exists $\mathbf{C}' \in \mathcal{C}$ such that $\max_{k\in[K]} \phi_k(\mathbf{C}') \leq -r$ for some $r > 0$. It then follows from part 1 that:

$$\begin{aligned}
\psi(\mathbf{C}^*) &= \min_{\mathbf{C}\in\mathcal{C},\,\boldsymbol{\xi}\in\Delta_d} \mathcal{L}^{\text{con}}(\mathbf{C},\boldsymbol{\xi},\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*) \\
&\leq \mathcal{L}^{\text{con}}(\mathbf{C}',\mathbf{C}',\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*) \\
&\leq \psi(\mathbf{C}') + \langle\boldsymbol{\mu}^*, \boldsymbol{\phi}(\mathbf{C}')\rangle \;=\; \psi(\mathbf{C}') - r\|\boldsymbol{\mu}^*\|_1.
\end{aligned}$$

We thus have:

$$\|\boldsymbol{\mu}^*\|_1 \;\leq\; (\psi(\mathbf{C}') - \psi(\mathbf{C}^*))/r \;=\; 1/r.$$

For part 4, letting $\omega(\boldsymbol{\xi}) = \psi(\boldsymbol{\xi}) + \langle\boldsymbol{\mu}^*, \boldsymbol{\phi}(\boldsymbol{\xi})\rangle$, we note that:

$$\begin{aligned}
\max_{\boldsymbol{\lambda}\in\mathbb{R}^d} \min_{\boldsymbol{\xi}\in\Delta_d} \mathcal{L}^{\text{con}}(\mathbf{C}^*,\boldsymbol{\xi},\boldsymbol{\lambda},\boldsymbol{\mu}^*) &= \min_{\boldsymbol{\xi}\in\Delta_d} \mathcal{L}^{\text{con}}(\mathbf{C}^*,\boldsymbol{\xi},\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*) \\
&= \min_{\boldsymbol{\xi}\in\Delta_d} \{\psi(\boldsymbol{\xi}) + \langle\boldsymbol{\mu}^*, \boldsymbol{\phi}(\boldsymbol{\xi})\rangle - \langle\boldsymbol{\lambda}^*, \boldsymbol{\xi}\rangle\} + \langle\boldsymbol{\lambda}^*, \mathbf{C}^*\rangle \\
&= \min_{\boldsymbol{\xi}\in\Delta_d} \{\omega(\boldsymbol{\xi}) - \langle\boldsymbol{\lambda}^*, \boldsymbol{\xi}\rangle\} + \langle\boldsymbol{\lambda}^*, \mathbf{C}^*\rangle \\
&= -\omega^*(\boldsymbol{\lambda}^*) + \langle\boldsymbol{\lambda}^*, \mathbf{C}^*\rangle, \qquad (21)
\end{aligned}$$

where ω^* denotes the Fenchel conjugate of ω. We similarly note that:

$$\begin{aligned}
\max_{\boldsymbol{\lambda}\in\mathbb{R}^d} \min_{\boldsymbol{\xi}\in\Delta_d} \mathcal{L}^{\text{con}}(\mathbf{C}^*,\boldsymbol{\xi},\boldsymbol{\lambda},\boldsymbol{\mu}^*) &= \max_{\boldsymbol{\lambda}\in\mathbb{R}^d} \left\{ \min_{\boldsymbol{\xi}\in\Delta_d} \{\omega(\boldsymbol{\xi}) - \langle\boldsymbol{\lambda}, \boldsymbol{\xi}\rangle\} + \langle\boldsymbol{\lambda}, \mathbf{C}^*\rangle \right\} \\
&= \max_{\boldsymbol{\lambda}\in\mathbb{R}^d} \{-\omega^*(\boldsymbol{\lambda}) + \langle\boldsymbol{\lambda}, \mathbf{C}^*\rangle\} \\
&= \omega^{**}(\mathbf{C}^*) \;=\; \omega(\mathbf{C}^*), \qquad (22)
\end{aligned}$$

where ω^{**} denotes the second Fenchel conjugate of ω. From (21) and (22), its clear that:

$$\omega(\mathbf{C}^*) \;=\; -\omega^*(\boldsymbol{\lambda}^*) + \langle \boldsymbol{\lambda}^*, \mathbf{C}^* \rangle.$$

An application of the Fenchel-Young inequality then gives us that:

$$\lambda^* = \nabla\omega(\mathbf{C}^*) = \nabla\psi(\mathbf{C}^*) + \sum_{k=1}^{K} \mu_k^* \nabla\phi_k(\mathbf{C}^*).$$

We can thus bound the norm of $\boldsymbol{\lambda}^*$ as:

$$\begin{aligned} \|\lambda^*\|_2 &\leq \|\nabla\psi(\mathbf{C}^*)\|_2 + \sum_{k=1}^{K} |\mu_k^*| \|\nabla\phi_k(\mathbf{C}^*)\|_2 \\ &\leq \|\nabla\psi(\mathbf{C}^*)\|_2 + \|\boldsymbol{\mu}^*\|_1 \max_{k\in K} \|\nabla\phi_k(\mathbf{C}^*)\|_2 \;=\; L(1+1/r), \end{aligned}$$

which follows from part 2 and the fact that ψ and ϕ_ks are Lipschitz w.r.t. the ℓ_1-norm. □

Proof of Theorem 17. We begin by writing out the Lagrangian from (2):

$$\mathcal{L}^{\text{con}}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \psi(\boldsymbol{\xi}) + \langle \boldsymbol{\lambda}, \mathbf{C} - \boldsymbol{\xi} \rangle + \langle \boldsymbol{\mu}, \boldsymbol{\phi}(\boldsymbol{\xi}) \rangle = \underbrace{\psi(\boldsymbol{\xi}) - \langle \boldsymbol{\lambda}, \boldsymbol{\xi} \rangle + \langle \boldsymbol{\mu}, \boldsymbol{\phi}(\mathbf{C}) \rangle}_{\mathcal{L}_1(\boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu})} + \underbrace{\langle \boldsymbol{\lambda}, \mathbf{C} \rangle}_{\mathcal{L}_2(\mathbf{C}, \boldsymbol{\lambda})}.$$

Optimality. To show optimality, note that $\mathcal{L}_1$ is convex in ξ and linear in $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, and $\mathcal{L}_2$ is linear both in $\mathbf{C}$ and $\boldsymbol{\lambda}$. The use of a (ρ, ρ', δ)-approximate LMO to compute $\mathbf{C}^t$ and h^t at each iteration gives us with probability at least $1 - \delta$ (over draw of S):

$$\begin{aligned} \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_2(\mathbf{C}^t, \boldsymbol{\lambda}^t) &\leq \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_2(\mathbf{C}[h^t], \boldsymbol{\lambda}^t) \;+\; \|\boldsymbol{\lambda}^t\|_1 \|\mathbf{C}^t - \mathbf{C}[h^t]\|_\infty \\ &\leq \|\boldsymbol{\lambda}^t\|_\infty \frac{1}{T}\sum_{t=1}^{T} \min_{\mathbf{C}\in\mathcal{C}} \left\langle \frac{\boldsymbol{\lambda}^t}{\|\boldsymbol{\lambda}^t\|_\infty}, \mathbf{C} \right\rangle \;+\; \|\boldsymbol{\lambda}^t\|_\infty \rho \;+\; \|\boldsymbol{\lambda}^t\|_1 \rho' \\ &\leq \min_{\mathbf{C}\in\mathcal{C}} \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_2(\mathbf{C}, \boldsymbol{\lambda}^t) \;+\; 2L(1+1/r)\rho + 2L\sqrt{d}(1+1/r)\rho'. \\ &= \min_{\mathbf{C}\in\mathcal{C}} \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_2(\mathbf{C}, \boldsymbol{\lambda}^t) \;+\; \bar{\rho}, \end{aligned} \tag{23}$$

where we denote $\bar{\rho} = 2L(1+1/r)\rho + 2L\sqrt{d}(1+1/r)\rho'$.

Next, we apply the classical regret bound guarantee for online gradient *descent* (Zinkevich, 2003; Shalev-Shwartz, 2011), we have from the sequence of objectives $\mathcal{L}_1(\cdot, \boldsymbol{\lambda}^t, \boldsymbol{\mu}^t)$'s (where the optimization is over ξ). Note that

$$\begin{aligned} \|\nabla_{\boldsymbol{\xi}} \mathcal{L}_1(\boldsymbol{\xi}, \boldsymbol{\lambda}^t, \boldsymbol{\mu}^t)\|_2 &\leq \|\nabla_{\boldsymbol{\xi}}\psi(\boldsymbol{\xi})\|_2 + \|\boldsymbol{\lambda}^t\|_2 + \|\boldsymbol{\mu}^t\|_1 \max_k \|\nabla_{\boldsymbol{\xi}}\phi_k(\boldsymbol{\xi})\|_2 \\ &\leq L + 2L(1+1/r) + 2L/r = (3+4/r)L \;\leq\; \bar{L}. \end{aligned}$$

Also note that $\max_{\boldsymbol{\xi} \in \Delta_d} \|\boldsymbol{\xi}\|_2 \leq 1$. So with $\eta = \frac{1}{\bar{L}\sqrt{2T}}$, we have:

$$\frac{1}{T} \sum_{t=1}^{T} \mathcal{L}_1(\boldsymbol{\xi}^t, \boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) \;\leq\; \min_{\boldsymbol{\xi} \in [0,1]^d} \sum_{t=1}^{T} \mathcal{L}_1(\boldsymbol{\xi}, \boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) \;+\; \frac{\sqrt{2}\bar{L}}{\sqrt{T}}. \tag{24}$$

Combining (23) and (24), we have with probability at least $1 - \delta$ (over draw of S):

$$\begin{aligned}
\frac{1}{T} \sum_{t=1}^{T} \mathcal{L}^{\text{con}}(\mathbf{C}^t, \boldsymbol{\xi}^t, \boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) \;&\leq\; \min_{\mathbf{C} \in \mathcal{C},\, \boldsymbol{\xi} \in [0,1]^d} \sum_{t=1}^{T} \mathcal{L}^{\text{con}}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) \;+\; \frac{\sqrt{2}\bar{L}}{\sqrt{T}} \;+\; \bar{\rho} \\
&=\; \min_{\mathbf{C} \in \mathcal{C},\, \boldsymbol{\xi} \in [0,1]^d} \mathcal{L}^{\text{con}}(\mathbf{C}, \boldsymbol{\xi}, \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}}) \;+\; \frac{\sqrt{2}\bar{L}}{\sqrt{T}} \;+\; \bar{\rho} \\
&\leq\; \max_{\boldsymbol{\lambda} \in \mathbb{R}^d, \boldsymbol{\mu} \in \mathbb{R}^K_+} \min_{\mathbf{C} \in \mathcal{C},\, \boldsymbol{\xi} \in [0,1]^d} \mathcal{L}^{\text{con}}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \;+\; \frac{\sqrt{2}\bar{L}}{\sqrt{T}} \;+\; \bar{\rho} \\
&=\; \min_{\mathbf{C} \in \mathcal{C}} \left\{ \max_{\boldsymbol{\lambda} \in \mathbb{R}^d, \boldsymbol{\mu} \in \mathbb{R}^K_+} \min_{\boldsymbol{\xi} \in [0,1]^d} \mathcal{L}^{\text{con}}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \right\} \;+\; \frac{\sqrt{2}\bar{L}}{\sqrt{T}} \;+\; \bar{\rho} \\
&=\; \min_{\mathbf{C} \in \mathcal{C}:\, \boldsymbol{\phi}(\mathbf{C}) \leq \mathbf{0}} \psi(\mathbf{C}) \;+\; \frac{\sqrt{2}\bar{L}}{\sqrt{T}} \;+\; \bar{\rho},
\end{aligned} \tag{25}$$

where in the second step $\bar{\boldsymbol{\lambda}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\lambda}^t$ and $\bar{\boldsymbol{\mu}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\mu}^t$ and we use the linearity of $\mathcal{L}$ in $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$; in the fourth step we use strong duality to interchange the max and min; and the last step follows from Lemma 33 (part 1).

Similarly, we apply the standard online gradient *ascent* analysis on the sequence of losses $\mathcal{L}^{\text{con}}(\mathbf{C}^t, \boldsymbol{\xi}^t, \cdot, \cdot)$'s, where the optimization is over $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$. Note that $\|\nabla_{\boldsymbol{\lambda},\boldsymbol{\mu}} \mathcal{L}^{\text{con}}(\mathbf{C}^t, \boldsymbol{\xi}^t, \boldsymbol{\lambda}^t, \boldsymbol{\mu}^t)\|_2 = \|\mathbf{C}^t - \boldsymbol{\xi}^t\|_2 + \|\boldsymbol{\phi}(\boldsymbol{\xi}^t)\|_2 \leq 1 + B_{\boldsymbol{\phi}}$ and $\left\| \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\mu} \end{bmatrix} \right\|_2 \leq 2L(1 + 1/r) + 2/r \leq \bar{L}$ (from Lemma 33, parts 3–4). So with $\eta' = \frac{\bar{L}}{(1+B_{\boldsymbol{\phi}})\sqrt{2T}}$, we have:

$$\begin{aligned}
&\frac{1}{T} \sum_{t=1}^{T} \mathcal{L}^{\text{con}}(\mathbf{C}^t, \boldsymbol{\xi}^t, \boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) \\
&\geq\; \max_{\boldsymbol{\lambda} \in \Lambda,\, \boldsymbol{\mu} \in \Xi} \sum_{t=1}^{T} \mathcal{L}^{\text{con}}(\mathbf{C}^t, \boldsymbol{\xi}^t, \boldsymbol{\lambda}, \boldsymbol{\mu}) \;-\; \frac{\sqrt{2}\bar{L}(1 + B_{\boldsymbol{\phi}})}{\sqrt{T}} \\
&\geq\; \max_{\boldsymbol{\lambda} \in \Lambda,\, \boldsymbol{\mu} \in \Xi} \left\{ \sum_{t=1}^{T} \mathcal{L}^{\text{con}}(\mathbf{C}[h^t], \boldsymbol{\xi}^t, \boldsymbol{\lambda}, \boldsymbol{\mu}) \;-\; \|\boldsymbol{\lambda}\|_1 \|\mathbf{C}^t - \mathbf{C}[h^t]\|_\infty \right\} \;-\; \frac{\sqrt{2}\bar{L}(1 + B_{\boldsymbol{\phi}})}{\sqrt{T}} \\
&\geq\; \max_{\boldsymbol{\lambda} \in \Lambda,\, \boldsymbol{\mu} \in \Xi} \sum_{t=1}^{T} \mathcal{L}^{\text{con}}(\mathbf{C}[h^t], \boldsymbol{\xi}^t, \boldsymbol{\lambda}, \boldsymbol{\mu}) \;-\; 2L(1 + 1/r)\sqrt{d}\rho' \;-\; \frac{\sqrt{2}\bar{L}(1 + B_{\boldsymbol{\phi}})}{\sqrt{T}} \\
&\geq\; \max_{\boldsymbol{\lambda} \in \Lambda,\, \boldsymbol{\mu} \in \Xi} \mathcal{L}^{\text{con}}(\mathbf{C}[\bar{h}], \bar{\boldsymbol{\xi}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \;-\; 2L(1 + 1/r)\sqrt{d}\rho' \;-\; \frac{\sqrt{2}\bar{L}(1 + B_{\boldsymbol{\phi}})}{\sqrt{T}} \\
&=\; \max_{\boldsymbol{\lambda} \in \Lambda} \left\{ \psi(\bar{\boldsymbol{\xi}}) + \langle \boldsymbol{\lambda}, \mathbf{C}[\bar{h}] - \bar{\boldsymbol{\xi}} \rangle \right\} \;+\; \max_{\boldsymbol{\mu} \in \Xi} \langle \boldsymbol{\mu}, \boldsymbol{\phi}(\bar{\boldsymbol{\xi}}) \rangle \;-\; 2L(1 + 1/r)\sqrt{d}\rho' \;-\; \frac{\sqrt{2}\bar{L}(1 + B_{\boldsymbol{\phi}})}{\sqrt{T}}
\end{aligned} \tag{26}$$

$$
\begin{aligned}
&\geq \min_{\boldsymbol{\xi}\in[0,1]^d}\left\{\max_{\boldsymbol{\lambda}\in\Lambda}\left\{\psi(\boldsymbol{\xi})+\langle\boldsymbol{\lambda},\mathbf{C}[\bar{h}]-\boldsymbol{\xi}\rangle\right\}+\max_{\boldsymbol{\mu}\in\Xi}\langle\boldsymbol{\mu},\boldsymbol{\phi}(\boldsymbol{\xi})\rangle\right\}-2L(1+1/r)\sqrt{d}\rho'-\frac{\sqrt{2}\bar{L}(1+B_{\boldsymbol{\phi}})}{\sqrt{T}}\\
&\geq \min_{\boldsymbol{\xi}\in[0,1]^d}\left\{\max_{\boldsymbol{\lambda}\in\Lambda}\left\{\psi(\boldsymbol{\xi})+\langle\boldsymbol{\lambda},\mathbf{C}[\bar{h}]-\boldsymbol{\xi}\rangle\right\}+\langle\mathbf{0},\boldsymbol{\phi}(\boldsymbol{\xi})\rangle\right\}-2L(1+1/r)\sqrt{d}\rho'-\frac{\sqrt{2}\bar{L}(1+B_{\boldsymbol{\phi}})}{\sqrt{T}}\\
&= \psi(\mathbf{C}[\bar{h}])-2L(1+1/r)\sqrt{d}\rho'-\frac{\sqrt{2}\bar{L}(1+B_{\boldsymbol{\phi}})}{\sqrt{T}},
\end{aligned}
\tag{27}
$$

where in the third step, we use the fact that for any $\lambda\in\Lambda$, $\|\boldsymbol{\lambda}\|_\infty\leq\|\boldsymbol{\lambda}\|_2\leq L$, and the property of the LMO. In the fourth step, we use $\mathbf{C}[\bar{h}]=\frac{1}{T}\sum_{t=1}^T\mathbf{C}[h^t]$ and $\bar{\boldsymbol{\xi}}=\frac{1}{T}\sum_{t=1}^T\boldsymbol{\xi}^t$, and use the linearity of $\mathcal{L}$ in $\mathbf{C}$, and convexity of $\mathcal{L}$ in $\boldsymbol{\xi}$ and Jensen's inequality. In the last step, we apply Lemma 33 (part 2). The last six steps hold with probability at least $1-\delta$.

Combining (25) and (27), we get with probability at least $1-\delta$ (over draw of S), for any $\boldsymbol{\mu}'\in\Xi$

$$
\psi(\mathbf{C}[\bar{h}])\leq\min_{\mathbf{C}\in\mathcal{C}:\,\boldsymbol{\phi}(\mathbf{C})\leq\mathbf{0}}\psi(\mathbf{C})+\frac{\sqrt{2}\bar{L}(2+B_{\boldsymbol{\phi}})}{\sqrt{T}}+2L(1+1/r)(\rho+2\sqrt{d}\rho').
$$

Setting $B_{\boldsymbol{\phi}}=\sqrt{K}$ and $T=(K+1)/\epsilon^2$ completes the proof of optimality.

Feasibility. Let $\mathbf{C}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*$ be as defined in Lemma 33. To show feasibility, combining (25) and (26), and interchanging the min and max, we get:

$$
\max_{\boldsymbol{\lambda}\in\Lambda,\boldsymbol{\mu}\in\Xi}\left\{\psi(\bar{\boldsymbol{\xi}})+\langle\boldsymbol{\lambda},\mathbf{C}[\bar{h}]-\bar{\boldsymbol{\xi}}\rangle+\langle\boldsymbol{\mu},\boldsymbol{\phi}(\bar{\boldsymbol{\xi}})\rangle\right\}\leq\psi(\mathbf{C}^*)+\widetilde{\rho}+\frac{\sqrt{2}\bar{L}(2+B_{\boldsymbol{\phi}})}{\sqrt{T}},
\tag{28}
$$

where we denote $\widetilde{\rho}=2L(1+1/r)(\rho+2\sqrt{d}\rho')$ Let $k'\in\operatorname{argmax}_{k\in[K]}\phi_k(\mathbf{C}[\bar{h}])$ denote the index of the most-violated among the K constraints $\phi_1(\mathbf{C}[\bar{h}]),\ldots,\phi_K(\mathbf{C}[\bar{h}])$. Also let $\boldsymbol{\lambda}'=\boldsymbol{\lambda}^*$ and $\mu'_{k'}=\mu^*_{k'}+\frac{1}{r}$ and $\mu'_k=\mu^*_k,\forall k\neq k'$. Note that $\boldsymbol{\lambda}'\in\Lambda$ and $\boldsymbol{\mu}'\in\Xi$. Substituting (μ',λ') into the LHS of (28), we have:

$$
\psi(\bar{\boldsymbol{\xi}})+\langle\boldsymbol{\lambda}^*,\mathbf{C}[\bar{h}]-\bar{\boldsymbol{\xi}}\rangle+\langle\boldsymbol{\mu}^*,\boldsymbol{\phi}(\boldsymbol{\xi})\rangle+\frac{1}{r}\max_k\phi_k(\bar{\boldsymbol{\xi}})\leq\psi(\mathbf{C}^*)+\widetilde{\rho}+\frac{\sqrt{2}\bar{L}(1+B_{\boldsymbol{\phi}})}{\sqrt{T}},
$$

and we further get:

$$
\begin{aligned}
&\min_{\mathbf{C}\in\mathcal{C},\,\boldsymbol{\xi}\in[0,1]^d}\left\{\psi(\boldsymbol{\xi})+\langle\boldsymbol{\lambda}^*,\mathbf{C}-\boldsymbol{\xi}\rangle+\langle\boldsymbol{\mu}^*,\boldsymbol{\phi}(\boldsymbol{\xi})\rangle\right\}+\frac{1}{r}\max_k\phi_k(\bar{\boldsymbol{\xi}})\\
&\quad\leq\psi(\mathbf{C}^*)+\widetilde{\rho}+\frac{\sqrt{2}\bar{L}(1+B_{\boldsymbol{\phi}})}{\sqrt{T}}.
\end{aligned}
$$

Applying Lemma 33 (part 1),

$$
\psi(\mathbf{C}^*)+\frac{1}{r}\max_k\phi_k(\bar{\boldsymbol{\xi}})\leq\psi(\mathbf{C}^*)+\widetilde{\rho}+\frac{\sqrt{2}\bar{L}(1+B_{\boldsymbol{\phi}})}{\sqrt{T}},
$$

giving us for all k:

$$
\phi_k(\bar{\boldsymbol{\xi}})\leq r\left(\widetilde{\rho}+\frac{\sqrt{2}\bar{L}(1+B_{\boldsymbol{\phi}})}{\sqrt{T}}\right).
\tag{29}
$$

Next set $\boldsymbol{\mu}' = \boldsymbol{\mu}^*$ and

$$\lambda'_{j'} = \lambda^*_{j'} + \frac{L(1+1/r)}{\left\|\mathbf{C}[\bar{h}] - \bar{\boldsymbol{\xi}}\right\|_2}(\mathbf{C}_{j'}[\bar{h}] - \bar{\xi}_{j'}).$$

Substituting (μ', λ') into the LHS of (28), we have:

$$\begin{aligned}\psi(\bar{\boldsymbol{\xi}}) + \langle \boldsymbol{\lambda}^*, \mathbf{C}[\bar{h}] - \bar{\boldsymbol{\xi}}\rangle \;+\; \langle \boldsymbol{\mu}^*, \boldsymbol{\phi}(\bar{\boldsymbol{\xi}})\rangle \;+\; L(1+1/r)\left\|\mathbf{C}[\bar{h}] - \bar{\boldsymbol{\xi}}\right\|_2 \\ \leq \quad \psi(\mathbf{C}^*) \;+\; \widetilde{\rho} \;+\; \frac{\sqrt{2}\bar{L}(1+B_{\boldsymbol{\phi}})}{\sqrt{T}},\end{aligned}$$

and again taking a min over $\mathbf{C}$ and $\boldsymbol{\xi}$ and applying Lemma 33, we get

$$\left\|\mathbf{C}[\bar{h}] - \bar{\boldsymbol{\xi}}\right\|_2 \quad \leq \quad \frac{1}{L(1+1/r)}\left(\widetilde{\rho} \;+\; \frac{\sqrt{2}\bar{L}(1+B_{\boldsymbol{\phi}})}{\sqrt{T}}\right). \tag{30}$$

Combining (29) and (30), and using the Lipschitz property of ϕ_k, we get for all k:

$$\begin{aligned}\phi_k(\mathbf{C}[\bar{h}]) \quad &\leq \quad L\left\|\mathbf{C}[\bar{h}] - \bar{\boldsymbol{\xi}}\right\|_2 \;+\; r\left(\widetilde{\rho} \;+\; \frac{\sqrt{2}\bar{L}(1+B_{\boldsymbol{\phi}})}{\sqrt{T}}\right) \\ &\leq \quad \frac{r(2+r)}{1+r}\left(\widetilde{\rho} \;+\; \frac{\sqrt{2}\bar{L}(1+B_{\boldsymbol{\phi}})}{\sqrt{T}}\right) \;\leq\; r\left(\widetilde{\rho} \;+\; \frac{\sqrt{2}\bar{L}(1+B_{\boldsymbol{\phi}})}{\sqrt{T}}\right).\end{aligned}$$

Setting $B_{\boldsymbol{\phi}} = 2\sqrt{K}$ and $T = (K+1)/\epsilon^2$ completes the proof of feasibility. □

A.10 Proof of Theorem 18 (Ellipsoid for Constrained Problems)

Theorem ((Restated) Convergence of ConEllipsoid)**.** *Fix $\epsilon \in (0,1)$. Let $\psi : [0,1]^d \to [0,1]$ and $\phi_1, \ldots, \phi_K : [0,1]^d \to [-1,1]$ be convex and L-Lipschitz w.r.t. the ℓ_2 norm. Let Ω in Algorithm 6 be a (ρ, ρ', δ)-approximate LMO for sample size N. Suppose the strict feasibility condition in Assumption 1 holds for radius $r > 0$. Let the initial classifier h^0 satisfy this condition, i.e. $\boldsymbol{\phi}(\mathbf{C}[h^0]) \leq -r$ and $\mathbf{C}[h^0] = \mathbf{C}^0$. Let $\bar{d} = d + K$. Let $\bar{h}$ be the classifier returned by Algorithm 7 when run for $T > 2\bar{d}^2 \log(\frac{\bar{d}}{\epsilon})$ iterations with initial radius $a > 2(L + \frac{L+1}{r})$. Then with probability $\geq 1 - \delta$ over draw of $S \sim D^N$, we have*

$$\begin{aligned}&\textbf{\textit{Optimality:}}\;\; \psi(\mathbf{C}[\bar{h}]) \;\leq \min_{\mathbf{C}\in\mathcal{C}:\, \phi_k(\mathbf{C})\leq 0, \forall k} \psi(\mathbf{C}) + \big(4a\big)\epsilon + 4a(\rho + 2\sqrt{d}\rho'); \\ &\textbf{\textit{Feasibility:}}\;\; \phi_k(\mathbf{C}[\bar{h}]) \;\leq\; a(\rho + 2\sqrt{d}\rho'),\; \forall k \in [K]\end{aligned}$$

In both the constrained and unconstrained versions of the Ellipsoid Algorithm, successive ellipsoids are constructed by obtaining the Löwner-John ellipsoid (JLE), i.e., the minimum volume ellipsoid containing the intersection of the current ellipsoid and a half space obtained by drawing a cutting hyperplane through the current center. This process yields a sequence of ellipsoids with geometrically decreasing volumes. We restate the lemma from Bubeck (2015) that establishes this fact.

Lemma 34. *Let the ellipsoid $\mathcal{E}^0 = \{\mathbf{x} \in \mathbb{R}^d : (\mathbf{x} - \mathbf{c}_0)^\top \mathbf{H}_0^{-1}(\mathbf{x} - \mathbf{c}_0) \leq 1)\}$, where $H_0 \in \mathbb{R}^{d \times d}$ is a positive definite matrix and $\mathbf{c}_0, \in \mathbb{R}^d$. Let $(\mathbf{H}, \mathbf{c}) = \text{JLE}(\mathbf{H}_0, \mathbf{c}_0, \mathbf{g})$, where* JLE *refers to the subroutine 3(a). Let the ellipsoid $\mathcal{E} = \{\mathbf{x} \in \mathbb{R}^d : (\mathbf{x} - \mathbf{c})^\top \mathbf{H}^{-1}(\mathbf{x} - \mathbf{c}) \leq 1)\}$. Then,*

$$\mathcal{E} \supset \mathcal{E}^0 \cap \{\mathbf{x} \in \mathbb{R}^d : \mathbf{g}^\top(\mathbf{x} - \mathbf{c}_0) \geq 0\}$$
$$\text{vol}(\mathcal{E}) \leq \exp\left(\frac{-1}{2d}\right) \text{vol}(\mathcal{E}^0)$$

where vol *refers to the standard d-dimensional volume.*

We will define some functions and variables below that will be useful in our proofs:

$$\begin{aligned}
\mathcal{L}^{\text{con}}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= \psi(\boldsymbol{\xi}) + \boldsymbol{\lambda}^\top(\mathbf{C} - \boldsymbol{\xi}) + \boldsymbol{\mu}^\top \boldsymbol{\phi}(\boldsymbol{\xi}) \\
f^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) &= \min_{\mathbf{C} \in \mathcal{C}, \boldsymbol{\xi} \in \Delta_d} \mathcal{L}^{\text{con}}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \\
\mathcal{R}^0 &:= \{\mathbf{x} \in \mathbb{R}^{d+K} : ||\mathbf{x}||_2 \leq a, \mathbf{x}_{d+i} \geq 0, \ \forall i \in \{1, 2, ..., K\}\} \\
\widehat{f}^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) &= f^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) - \infty \mathbf{1}([\boldsymbol{\lambda}, \boldsymbol{\mu}] \notin \mathcal{R}^0) \\
\boldsymbol{\xi}(\boldsymbol{\lambda}, \boldsymbol{\mu}) &\in \text{argmin}_{\boldsymbol{\xi} \in \Delta_d} \psi(\boldsymbol{\xi}) - \boldsymbol{\lambda}^\top \boldsymbol{\xi} + \boldsymbol{\mu}^\top \boldsymbol{\phi}(\boldsymbol{\xi})
\end{aligned}$$

The helper function $\widehat{f}^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu})$ is equal to $f^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu})$ when $[\boldsymbol{\lambda}, \boldsymbol{\mu}] \in \mathcal{R}^0$. Let h^t and $\mathbf{C}^t$ be the iterates in Algorithm 7. Let $\mathcal{E}^t$ denote the ellipsoid centered at $[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]$ with axes given by the eigen vectors of $\mathbf{A}^t$, with axes lengths squared given by the corresponding eigenvalues of $\mathbf{A}^t$, i.e.

$$\mathcal{E}^t = \{[\boldsymbol{\lambda}, \boldsymbol{\mu}] \in \mathbb{R}^{d+K} : [\boldsymbol{\lambda} - \boldsymbol{\lambda}^t, \boldsymbol{\mu} - \boldsymbol{\mu}^t]^\top (\mathbf{A}^t)^{-1} [\boldsymbol{\lambda} - \boldsymbol{\lambda}^t, \boldsymbol{\mu} - \boldsymbol{\mu}^t] \leq 1\}$$

We abuse notation sometimes in the proof below by interchangeably using the ellipsoid $\mathcal{E}^t$ and its corresponding center, axis matrix $[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t], \mathbf{A}^t$ whenever the context is clear. For example, line 14 of Algorithm 7 can be written compactly as $\mathcal{E}^{t+1} = \text{JLE}(\mathcal{E}^t, [\mathbf{C}^t - \boldsymbol{\xi}^t, \boldsymbol{\phi}(\boldsymbol{\xi}^t)])$.

A.10.1 BOUNDING THE DUAL SUBOPTIMALITY OF $[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]$

We first prove, that for any iteration $t \in \{0, 1, ..., T-1\}$, if $[\boldsymbol{\lambda}, \boldsymbol{\mu}] \notin \mathcal{R}^0$, then $\mathcal{E}^{t+1} \supseteq \{\mathcal{E}^t \cap \mathcal{R}^0\}$. We establish this in the following three lemmas.

Lemma 35. *If at any iteration $t \in \{0, 1, ..., T-1\}$, $|||[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]||_2 > a$, then $\mathcal{E}^{t+1} \supseteq \{\mathcal{E}^t \cap \mathcal{R}^0\}$*

Proof. Let $t \in \{0, 1, ..., T-1\}$, such that, $|||[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]||_2 > a$. In such a case, the **if** condition (line 5) of algorithm 7 gets invoked and we obtain the new ellipsoid $\mathcal{E}^{t+1}$. Due to the JLE construction, we get that

$$\begin{aligned}
\mathcal{E}^{t+1} &\supseteq \mathcal{E}^t \cap \{\mathbf{x} \in \mathbb{R}^{d+K} : (\mathbf{x} - [\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t])^\top (-[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t])) \geq 0\} \\
&= \mathcal{E}^t \cap \{\mathbf{x} \in \mathbb{R}^{d+K} : \mathbf{x}^\top [\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t] \leq |||[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]||_2^2]\} \\
&\supseteq \mathcal{E}^t \cap \mathbf{B}(\mathbf{0}_{d+K}, |||[\widehat{\boldsymbol{\lambda}}^t, \widehat{\boldsymbol{\mu}}^t]||_2) \supseteq \{\mathcal{E}^t \cap \mathcal{R}^0\}
\end{aligned}$$

Thus, $\mathcal{E}^{t+1} \supseteq \{\mathcal{E}^t \cap \mathcal{R}^0\}$. □

Lemma 36. *If at any iteration $t \in \{0, 1, ..., T-1\}$, $|||[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]||_2 \leq a$, and $\boldsymbol{\mu}^t \not\geq \mathbf{0}$, then $\mathcal{E}^{t+1} \supseteq \{\mathcal{E}^t \cap \mathcal{R}^0\}$*

Proof. Let $t \in \{0, 1, ..., T-1\}$, such that, $|||[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]||_2 \leq a$, while $\boldsymbol{\mu}^t \not\geq \mathbf{0}$. In such a case, the **else-if** condition (line 8) of algorithm 7 gets invoked and we obtain the new ellipsoid $\mathcal{E}^{t+1}$. Due to the JLE construction, we get that

$$\begin{aligned}\mathcal{E}^{t+1} &\supseteq \mathcal{E}^t \cap \{\mathbf{x} \in \mathbb{R}^{d+K} : (\mathbf{x} - [\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t])^\top ([\mathbf{0}_d, \text{pos}(-\boldsymbol{\mu}^t)])) \geq 0\} \\ &= \mathcal{E}^t \cap \{\mathbf{x} \in \mathbb{R}^{d+K} : \mathbf{x}^\top [\mathbf{0}_d, \text{pos}(-\boldsymbol{\mu}^t)] \geq [\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]^\top [\mathbf{0}_d, \text{pos}(-\boldsymbol{\mu}^t)] \\ &\supseteq \mathcal{E}^t \cap \{\mathbf{x} \in \mathbb{R}^{d+K} : \mathbf{x}_{d+i} \geq 0, \forall i \in 1, 2, ..., K\} \supseteq \{\mathcal{E}^t \cap \mathcal{R}^0\}\end{aligned}$$

Thus, $\mathcal{E}^{t+1} \supseteq \{\mathcal{E}^t \cap \mathcal{R}^0\}$. □

Lemma 37. *For any iteration $t \in \{0, 1, ..., T-1\}$ of Algorithm 7, if $[\boldsymbol{\lambda}, \boldsymbol{\mu}] \notin \mathcal{R}^0$, then $\mathcal{E}^{t+1} \supseteq \{\mathcal{E}^t \cap \mathcal{R}^0\}$*

Proof. The result follows directly from Lemmas 35 and 36. □

We would also like to prove that the optimal solution, i.e., the maximizer of f^{con} over $\mathbb{R}^d \times \mathbb{R}^K_+$ indeed lies inside our search space. In our setting, we show in 38 that the maximizer lies inside $\mathcal{R}^0$

Lemma 38. *Let $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ be a maximizer of f^{con} over $\mathbb{R}^d \times \mathbb{R}^K_+$. Then $[\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*] \in \mathcal{R}^0$*

Proof. From Lemma 33 (parts 3– 4) we have that $\|[\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*]\|_2 \leq L + \frac{L+1}{r} \leq a/2$. Thus:

$$\max_{\boldsymbol{\lambda} \in \mathbb{R}^d, \boldsymbol{\mu} \in \mathbb{R}^K_+} f^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \max_{\boldsymbol{\lambda} \in \mathbb{R}^d, \boldsymbol{\mu} \in \mathbb{R}^K_+} \widehat{f}^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \psi(\mathbf{C}^*) = \min_{\mathbf{C} \in \mathcal{C}, \boldsymbol{\phi}(\mathbf{C}) \leq 0} \psi(\mathbf{C})$$

This ensures that $[\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*] \in \mathcal{R}^0$ □

Lemmas 37 and 38 allow us to establish Lemma 39, which will be required in proving Theorem 44.

Lemma 39. *Let $\epsilon \in [0, 1]$ and $[\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*]$ be any maximiser of $\widehat{f}^{\text{con}}$. Define the convex set $\mathcal{R}^0_\epsilon \subseteq \mathcal{R}^0 \subseteq \mathbb{R}^d \times \mathbb{R}^K_+$ as*

$$\mathcal{R}^0_\epsilon := \{[\boldsymbol{\lambda}, \boldsymbol{\mu}] \in \mathcal{R}^0 : (1-\epsilon)[\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*] + \epsilon[\boldsymbol{\lambda}, \boldsymbol{\mu}]\}.$$

Let the number of iterations T in Algorithm 7, be such that $T > 2(d+K)^2 \log\left(\frac{2}{\epsilon}\right)$. Then there exists an iteration $t^* \in \{0, 1, .., T-1\}$ such that $\mathcal{R}^0_\epsilon \subseteq \mathcal{E}^{t^*}$ and $\mathcal{R}^0_\epsilon \not\subseteq \mathcal{E}^{t^*+1}$ and $[\boldsymbol{\lambda}^{t^*}, \boldsymbol{\mu}^{t^*}] \in \mathcal{R}^0$.

Proof. From Lemma 38, $[\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*] \in \mathcal{R}^0$ and thus $\mathcal{R}^0_\epsilon \subseteq \mathcal{R}^0 \subseteq \mathcal{E}^0$. We also have the following by simple geometry and the classic ellipsoid volume reduction result of Lemma 34.

$$\begin{aligned}\text{vol}(\mathcal{R}^0_\epsilon) &= \epsilon^{d+K} \text{vol}(\mathcal{R}^0) = \epsilon^{d+K} 2^{-K} \text{vol}(\mathcal{E}^0) \\ \text{vol}(\mathcal{E}^T) &\leq \exp\left(\frac{-T}{2(d+K)}\right) \text{vol}(\mathcal{E}^0) \\ &\leq \exp\left((d+K)\log\left(\frac{\epsilon}{2}\right)\right) \text{vol}(\mathcal{E}^0) < \text{vol}(\mathcal{R}^0_\epsilon)\end{aligned}$$

And hence $\mathcal{R}^0_\epsilon \not\subseteq \mathcal{E}^T$. Clearly, there exists an iteration $t^* \in \{0, 1, \ldots, T-1\}$ such that $\mathcal{R}^0_\epsilon \subseteq \mathcal{E}^{t^*}$ but $\mathcal{R}^0_\epsilon \not\subseteq \mathcal{E}^{t^*+1}$. If $[\boldsymbol{\lambda}^{t^*}, \boldsymbol{\mu}^{t^*}] \notin \mathcal{R}^0$, then by Lemma 37 we have that $\mathcal{E}^{t^*+1} \supseteq \mathcal{E}^{t^*} \cap \mathcal{R}^0 \supseteq \mathcal{R}^0_\epsilon$, giving a contradiction. Thus $[\boldsymbol{\lambda}^{t^*}, \boldsymbol{\mu}^{t^*}] \in \mathcal{R}^0$.

□

We now prove that f^{con} is a Lipschitz function w.r.t. ℓ_2 norm over the domain $\mathcal{R}^0$. We will exploit this fact later in the proof for Theorem 44.

Lemma 40. *f^{con} is a $\sqrt{d+K}$-Lipschitz function w.r.t. ℓ_2 norm over the domain $\mathcal{R}^0$.*

Proof. The difference f^{con} at $[\boldsymbol{\lambda}, \boldsymbol{\mu}] \in \mathcal{R}^0$ and $[\boldsymbol{\lambda}', \boldsymbol{\mu}'] \in \mathcal{R}^0$ can be bounded by:

$$\begin{aligned}
f^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) - f^{\text{con}}(\boldsymbol{\lambda}', \boldsymbol{\mu}') &= \min_{\mathbf{C} \in \mathcal{C}, \boldsymbol{\xi} \in \Delta_d} \mathcal{L}^{\text{con}}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) - \min_{\mathbf{C} \in \mathcal{C}, \boldsymbol{\xi} \in \Delta_d} \mathcal{L}^{\text{con}}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}', \boldsymbol{\mu}') \\
&\leq \max_{\mathbf{C} \in \mathcal{C}, \boldsymbol{\xi} \in \Delta_d} \left(\mathcal{L}^{\text{con}}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) - \mathcal{L}^{\text{con}}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}', \boldsymbol{\mu}') \right) \\
&\leq \max_{\mathbf{C} \in \mathcal{C}, \boldsymbol{\xi} \in \Delta_d} \Big((\boldsymbol{\lambda} - \boldsymbol{\lambda}')^\top (\mathbf{C} - \boldsymbol{\xi}) + (\boldsymbol{\mu} - \boldsymbol{\mu}')^\top \boldsymbol{\phi}(\boldsymbol{\xi}) \Big) \\
&\leq \max_{\mathbf{C} \in \mathcal{C}, \boldsymbol{\xi} \in \Delta_d} \left(\|\boldsymbol{\lambda} - \boldsymbol{\lambda}'\|_1 \|\mathbf{C} - \boldsymbol{\xi}\|_\infty + \|\boldsymbol{\mu} - \boldsymbol{\mu}'\|_1 \|\boldsymbol{\phi}(\boldsymbol{\xi})\|_\infty \right) \\
&\leq \|\boldsymbol{\lambda} - \boldsymbol{\lambda}'\|_1 + \|\boldsymbol{\mu} - \boldsymbol{\mu}'\|_1 = \|[\boldsymbol{\lambda}, \boldsymbol{\mu}] - [\boldsymbol{\lambda}', \boldsymbol{\mu}']\|_1 \\
&\leq \sqrt{d+K}\ \|[\boldsymbol{\lambda}, \boldsymbol{\mu}] - [\boldsymbol{\lambda}', \boldsymbol{\mu}']\|_2
\end{aligned}$$

Identically, $f^{\text{con}}(\boldsymbol{\lambda}', \boldsymbol{\mu}') - f^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) \leq \sqrt{d+K}\ \|[\boldsymbol{\lambda}, \boldsymbol{\mu}] - [\boldsymbol{\lambda}', \boldsymbol{\mu}']\|_2$. Thus $|f^{\text{con}}(\boldsymbol{\lambda}', \boldsymbol{\mu}') - f^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu})| \leq \sqrt{d+K}\ \|[\boldsymbol{\lambda}, \boldsymbol{\mu}] - [\boldsymbol{\lambda}', \boldsymbol{\mu}']\|_2$ which concludes the proof. □

Recall that we only have access to (ρ, ρ', δ)-approximate LMO. The sample and approximation errors induced by calls to this approximate LMO must be accounted for. It turns out, that despite having access to only an approximate LMO, we are able to achieve a desirable sub-optimality with probability $1 - \delta$ over the draw of random sample $S \sim D^N$. **The rest of the analysis will only apply for this high probability event**. We now present two lemmas that will be helpful in allowing us to show provided an approximate LMO, the iterates $[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]$ approximately maximize f^{con} and subsequently, we will use these results to convert our dual guarantees into primal guarantees.

Lemma 41. *Let $t \in \{0, 1, ..., T-1\}$. Then with probability $1-\delta$ (over draw of $S \sim D^N$) uniformly for all t, such that $[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t] \in \mathcal{R}^0$, we have that:*

- $\boldsymbol{\lambda}^{t\top} \mathbf{C}[h^t] \leq \min_{\mathbf{C} \in \mathcal{C}} \boldsymbol{\lambda}^{t\top} \mathbf{C} + a\rho$
- $\|\mathbf{C}[h^t] - \mathbf{C}^t\|_2 \leq \sqrt{d}\rho'$
- $\boldsymbol{\lambda}^{t\top} \mathbf{C}^t \leq \min_{\mathbf{C} \in \mathcal{C}} \boldsymbol{\lambda}^{t\top} \mathbf{C} + a(\rho + \sqrt{d}\rho')$

Proof. The first two inequalities are simply restatements of the definition of (ρ, ρ', δ)-approximate LMO. And the third follows by putting the first two together.

□

Lemma 42. *Let $t \in \{0, 1, ..., T-1\}$ and let $[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t] \in \mathcal{R}^0$. Then, $[\mathbf{C}^t - \boldsymbol{\xi}^t, \boldsymbol{\phi}(\boldsymbol{\xi}^t)]$ is a τ-supergradient to $\widehat{f}^{\text{con}}$ at $[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t] \in \mathcal{R}^0$, with $\tau = a(\rho + 2\sqrt{d}\rho')$, i.e. for all $\boldsymbol{\lambda} \in \mathbb{R}^d, \boldsymbol{\mu} \in \mathbb{R}^K$,*

$$\widehat{f}^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) \leq \widehat{f}^{\text{con}}(\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) + (\boldsymbol{\lambda} - \boldsymbol{\lambda}^t)^\top (\mathbf{C}^t - \boldsymbol{\xi}^t) + (\boldsymbol{\mu} - \boldsymbol{\mu}^t)^\top (\boldsymbol{\phi}(\boldsymbol{\xi}^t)) + \tau$$

Proof. Fix $[\boldsymbol{\lambda}, \boldsymbol{\mu}] \in \mathcal{R}^0$. We have that,

$$\begin{aligned}
\widehat{f}^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) &= \min_{\mathbf{C} \in \mathcal{C}, \boldsymbol{\xi} \in \Delta_d} \mathcal{L}(\mathbf{C}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \\
&\leq \mathcal{L}(\mathbf{C}[h^t], \boldsymbol{\xi}^t, \boldsymbol{\lambda}, \boldsymbol{\mu}) \\
&= \mathcal{L}(\mathbf{C}^t, \boldsymbol{\xi}^t, \boldsymbol{\lambda}, \boldsymbol{\mu}) + (\mathbf{C}[h^t] - \mathbf{C}^t)^\top \boldsymbol{\lambda} \\
&\leq \mathcal{L}(\mathbf{C}^t, \boldsymbol{\xi}^t, \boldsymbol{\lambda}, \boldsymbol{\mu}) + \|\mathbf{C}[h^t] - \mathbf{C}^t\|_2 \|\boldsymbol{\lambda}\|_2 \\
&\leq \mathcal{L}(\mathbf{C}^t, \boldsymbol{\xi}^t, \boldsymbol{\lambda}, \boldsymbol{\mu}) + a\sqrt{d}\rho'.
\end{aligned}$$

Further,

$$\begin{aligned}
\widehat{f}^{\text{con}}(\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) &= \min_{\mathbf{C} \in \mathcal{C}} {\boldsymbol{\lambda}^t}^\top \mathbf{C} + \min_{\boldsymbol{\xi} \in \Delta_d} \psi(\boldsymbol{\xi}) - {\boldsymbol{\lambda}^t}^\top \boldsymbol{\xi} + {\boldsymbol{\mu}^t}^\top \boldsymbol{\phi}(\boldsymbol{\xi}) \\
&\geq {\boldsymbol{\lambda}^t}^\top \mathbf{C}^t - a(\rho + \sqrt{d}\rho') + \psi(\boldsymbol{\xi}^t) - {\boldsymbol{\lambda}^t}^\top \boldsymbol{\xi}^t + {\boldsymbol{\mu}^t}^\top \boldsymbol{\phi}(\boldsymbol{\xi}^t) \\
&= \mathcal{L}(\mathbf{C}^t, \boldsymbol{\xi}^t, \boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) - a(\rho + \sqrt{d}\rho') \\
&= \mathcal{L}(\mathbf{C}^t, \boldsymbol{\xi}^t, \boldsymbol{\lambda}, \boldsymbol{\mu}) + (\boldsymbol{\lambda}^t - \boldsymbol{\lambda})^\top (\mathbf{C}^t - \boldsymbol{\xi}^t) + (\boldsymbol{\mu}^t - \boldsymbol{\mu})^\top \boldsymbol{\phi}(\boldsymbol{\xi}^t) - a(\rho + \sqrt{d}\rho') \\
&\geq \widehat{f}^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) - a\sqrt{d}\rho' + (\boldsymbol{\lambda}^t - \boldsymbol{\lambda})^\top (\mathbf{C}^t - \boldsymbol{\xi}^t) + (\boldsymbol{\mu}^t - \boldsymbol{\mu})^\top \boldsymbol{\phi}(\boldsymbol{\xi}^t) - a(\rho + \sqrt{d}\rho'),
\end{aligned}$$

as desired. If $[\boldsymbol{\lambda}, \boldsymbol{\mu}] \notin \mathcal{R}^0$, the result follows trivially. □

Equipped with lemmas 39, 42 and 40, we are now ready to prove that Algorithm 7 approximately maximizes f^{con}. The monograph by Bubeck (2015) presents a proof to derive the sub-optimality of the regular ellipsoid algorithm, where perfect (sub/ super) gradient access is assumed. In our setting, we only have access to approximate super-gradients. We show how to adapt the proof of Bubeck (2015) to our setting, in the proof for Theorem 44.

Lemma 43. *Let $\tau = a(\rho + 2\sqrt{d}\rho')$. For any $t \in \{0, 1, \ldots, T-1\}$, such that, $[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t] \in \mathcal{R}^0$*

$$\mathcal{E}^t \setminus \mathcal{E}^{t+1} \subset \{[\boldsymbol{\lambda}, \boldsymbol{\mu}] \in \mathbb{R}^{d+K} : \widehat{f}^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) \leq \widehat{f}^{\text{con}}(\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) + \tau\}$$

Proof. Pick $t \in \{0, 1, \ldots, T-1\}$, such that $[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t] \in \mathcal{R}^0$. We know by lemma 42 that $\mathbf{g}^t := [\mathbf{C}^t - \boldsymbol{\xi}^t, \boldsymbol{\phi}(\boldsymbol{\xi}^t)]$ is a τ super-gradient to $\widehat{f}^{\text{con}}$ at $[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]$. Thus, $\forall\, \boldsymbol{\lambda} \in \mathbb{R}^d,\ \forall\, \boldsymbol{\mu} \in \mathbb{R}^K$, we have that

$$\widehat{f}^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) \leq \widehat{f}^{\text{con}}(\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) + (g^t)^\top ([\boldsymbol{\lambda}, \boldsymbol{\mu}] - [\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]) + \tau \tag{31}$$

Since $[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t] \in \mathcal{R}^0$, the **else** condition (line 11) of Algorithm 7 gets invoked and we get that $\mathcal{E}^{t+1} = \text{JLE}(\mathcal{E}^t, g^t)$ and thus by Lemma 34 and Equation (31) we have the following:

$$\begin{aligned}
\mathcal{E}^{t+1} &\supseteq \mathcal{E}^t \cap \{[\boldsymbol{\lambda}, \boldsymbol{\mu}] \in \mathbb{R}^{d+K} : (\mathbf{g}^t)^\top ([\boldsymbol{\lambda}, \boldsymbol{\mu}] - [\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]) \geq 0\} \\
\mathcal{E}^t \setminus \mathcal{E}^{t+1} &\subseteq \{[\boldsymbol{\lambda}, \boldsymbol{\mu}] \in \mathbb{R}^{d+K} : (\mathbf{g}^t)^\top ([\boldsymbol{\lambda}, \boldsymbol{\mu}] - [\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]) < 0\} \\
&\subseteq \{[\boldsymbol{\lambda}, \boldsymbol{\mu}] \in \mathbb{R}^{d+K} : \widehat{f}^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) \leq \widehat{f}^{\text{con}}(\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) + \tau\}
\end{aligned}$$

where the second line follows from the argument that for any sets A, B, C, if $A \supset B \cap C$ then $B \setminus A \subseteq C^c$, and the last line follows from Equation (31). □

Theorem 44. *Let the assumptions stated in Theorem 18 hold. Then,*

$$\max_{0\leq t\leq T-1} \widehat{f}^{\text{con}}(\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) \geq \max_{\boldsymbol{\lambda},\boldsymbol{\mu}} \widehat{f}^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) - \left(4a\sqrt{d+K}\right) \cdot \exp\left(\frac{-T}{2(d+K)^2}\right) - \tau$$

where $\tau = a(\rho + 2\sqrt{d}\rho')$

Proof. Due to lemma 38, we know that $\exists\ [\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*] \in \mathcal{R}^0$, where $[\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*]$ is a maximizer of f^{con} over $\mathbb{R}^d \times \mathbb{R}^K_+$. Set $\epsilon = 2\exp\left(\frac{-T}{2(d+K)^2}\right)$ which implies $T > 2(d+K)^2 log(\frac{2}{\epsilon})$. Let $\mathcal{R}^0_\epsilon \subseteq \mathcal{R}^0 \subseteq \mathbb{R}^d \times \mathbb{R}^K_+$ be

$$\mathcal{R}^0_\epsilon := \{[\boldsymbol{\lambda}, \boldsymbol{\mu}] \in \mathcal{R}^0 : (1-\epsilon)[\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*] + \epsilon[\boldsymbol{\lambda}, \boldsymbol{\mu}]\}.$$

By Lemma 39, there exists an iteration $t^* \in \{0, 1, \ldots, T-1\}$, such that, $\mathcal{R}^0_\epsilon \subseteq \mathcal{E}^{t^*},\ \mathcal{R}^0_\epsilon \nsubseteq \mathcal{E}^{t^*+1}$ and $[\boldsymbol{\lambda}^{t^*}, \boldsymbol{\mu}^{t^*}] \in \mathcal{R}^0$. Pick any element $[\boldsymbol{\lambda}_\epsilon, \boldsymbol{\mu}_\epsilon] \in \mathcal{R}^0_\epsilon \setminus \mathcal{E}^{t^*+1} \subseteq \mathcal{E}^{t^*} \setminus \mathcal{E}^{t^*+1}$. Because of the definition of $\mathcal{R}^0_\epsilon,\ \exists\ [\boldsymbol{\lambda}^{'}, \boldsymbol{\mu}'] \in \mathcal{R}^0$, such that, $[\boldsymbol{\lambda}_\epsilon, \boldsymbol{\mu}_\epsilon] = (1-\epsilon)[\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*] + \epsilon[\boldsymbol{\lambda}^{'}, \boldsymbol{\mu}']$. Due to Lemma 43, we have that,

$$\begin{aligned}
\widehat{f}^{\text{con}}(\boldsymbol{\lambda}^{t^*}, \boldsymbol{\mu}^{t^*}) &\geq \widehat{f}^{\text{con}}(\boldsymbol{\lambda}_\epsilon, \boldsymbol{\mu}_\epsilon) - \tau \\
&= f^{\text{con}}(\boldsymbol{\lambda}_\epsilon, \boldsymbol{\mu}_\epsilon) - \tau \\
&= f^{\text{con}}((1-\epsilon)\boldsymbol{\lambda}^* + \epsilon\boldsymbol{\lambda}^{'}, (1-\epsilon)\boldsymbol{\mu}^* + \epsilon\boldsymbol{\mu}^{'}) - \tau \\
&\geq (1-\epsilon) f^{\text{con}}(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) + \epsilon f^{\text{con}}(\boldsymbol{\lambda}^{'}, \boldsymbol{\mu}^{'}) - \tau \\
&\geq (1-\epsilon) f^{\text{con}}(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) + \epsilon(f^{\text{con}}(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) - 2a\sqrt{d+K}) - \tau \\
&= f^{\text{con}}(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) - \epsilon(2a\sqrt{d+K}) - \tau \\
&= f^{\text{con}}(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) - 4a\sqrt{d+K}\exp\left(\frac{-T}{2(d+K)^2}\right) - \tau
\end{aligned}$$

the second inequality in the above argument is due to the concavity of f^{con} and the third inequality is due to the $\sqrt{d+K}$ Lipschitzness of f^{con} in $\mathcal{R}^0$ (Lemma 40) and the ℓ_2-norm diameter of the set $\mathcal{R}^0$ being bounded above by $2a$. The theorem follows from the equality of f^{con} and $\widehat{f}^{\text{con}}$ within $\mathcal{R}^0$. □

A.10.2 CONVERTING GUARANTEE ON $\widehat{f}^{\text{con}}$ TO PRIMAL OPTIMALITY-FEASIBILITY GUARANTEES

Now, we can bound the primal sub-optimality using a standard technique from optimization theory Lee et al. (2015). Throughout, we will appeal to the high-probability inequalities established in 41.

Lemma 45. *Denote* $\widetilde{\mathcal{C}} = \text{conv}(\{\mathbf{C}[h^0], \mathbf{C}[h^1], \ldots, \mathbf{C}[h^{T-1}]\})$. *We then have:*

$$\min_{\mathbf{C}\in\widetilde{\mathcal{C}}, \boldsymbol{\phi}(\mathbf{C})\leq 0} \psi(\mathbf{C}) \leq \min_{\mathbf{C}\in\mathcal{C}, \boldsymbol{\phi}(\mathbf{C})\leq 0} \psi(\mathbf{C}) + \left(4a\sqrt{d+K}\right) \cdot \exp\left(\frac{-T}{2(d+K)^2}\right) + 2\tau$$

where $\tau = a(\rho + 2\sqrt{d}\rho')$.

Proof. Consider an alternative version of f^{con} defined as

$$\widetilde{f}^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \min_{\mathbf{C}\in\widetilde{\mathbf{C}}, \boldsymbol{\xi}\in[0,1]^d} \mathcal{L}^{\text{con}}(\mathbf{C}, \xi, \boldsymbol{\lambda}, \boldsymbol{\mu}).$$

And let $\widehat{\widetilde{f}}^{\text{con}}$ be equal to $\widetilde{f}^{\text{con}}$ if its argument $\boldsymbol{\lambda}, \boldsymbol{\mu}$ is inside the ℓ_2-norm ball of radius a and $\boldsymbol{\mu} \geq 0$, and negative infinity otherwise.

Clearly we have that $\widehat{\widetilde{f}}^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) \geq \widehat{f}^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu})$. We can also show $\widehat{\widetilde{f}}^{\text{con}}$ and $\widehat{f}^{\text{con}}$ are close at the iterates $[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t]$. If $[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t] \notin \mathcal{R}^0$, then both sides are trivially equal to negative infinity. Suppose $[\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t] \in \mathcal{R}^0$, we then have:

$$\begin{aligned}
\widehat{\widetilde{f}}^{\text{con}}(\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) = \widetilde{f}^{\text{con}}(\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) &\leq \mathcal{L}(\mathbf{C}[h^t], \boldsymbol{\xi}^t, \boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) \\
&= \psi(\boldsymbol{\xi}^t) - {\boldsymbol{\lambda}^t}^\top \boldsymbol{\xi}^t + {\boldsymbol{\lambda}^t}^\top \mathbf{C}[h^t] + {\boldsymbol{\mu}^t}^\top \boldsymbol{\phi}(\boldsymbol{\xi}^t) \\
&= \min_{\boldsymbol{\xi} \in \Delta_d} \Big(\psi(\boldsymbol{\xi}) - {\boldsymbol{\lambda}^t}^\top \boldsymbol{\xi} + {\boldsymbol{\mu}^t}^\top \boldsymbol{\phi}(\boldsymbol{\xi}^t) \Big) + {\boldsymbol{\lambda}^t}^\top \mathbf{C}[h^t] \\
&\leq \min_{\boldsymbol{\xi} \in \Delta_d} \Big(\psi(\boldsymbol{\xi}) - {\boldsymbol{\lambda}^t}^\top \boldsymbol{\xi} + {\boldsymbol{\mu}^t}^\top \boldsymbol{\phi}(\boldsymbol{\xi}^t) \Big) + \min_{\mathbf{C} \in \mathcal{C}} {\boldsymbol{\lambda}^t}^\top \mathbf{C} + a\rho \\
&= \min_{\boldsymbol{\xi} \in \Delta_d, \mathbf{C} \in \mathcal{C}} \Big(\psi(\boldsymbol{\xi}) + {\boldsymbol{\lambda}^t}^\top (\mathbf{C} - \boldsymbol{\xi}) + {\boldsymbol{\mu}^t}^\top \boldsymbol{\phi}(\boldsymbol{\xi}) \Big) + a\rho \\
&= f^{\text{con}}(\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) + a\rho = \widehat{f}^{\text{con}}(\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) + a\rho. \qquad (32)
\end{aligned}$$

From Lemma 38 and the min-max theorem, we have the following:

$$\begin{aligned}
\max_{\boldsymbol{\lambda} \in \mathbb{R}^d, \boldsymbol{\mu} \in \mathbb{R}^K_+} \widehat{\widetilde{f}}^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) &= \max_{\boldsymbol{\lambda} \in \mathbb{R}^d, \boldsymbol{\mu} \in \mathbb{R}^K_+} \widetilde{f}^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) \\
&= \max_{\boldsymbol{\lambda} \in \mathbb{R}^d, \boldsymbol{\mu} \in \mathbb{R}^K_+} \min_{\mathbf{C} \in \widetilde{\mathcal{C}}, \boldsymbol{\xi} \in \Delta_d} \mathcal{L}^{\text{con}}(\mathbf{C}, \xi, \boldsymbol{\lambda}, \boldsymbol{\mu}) \\
&= \min_{\mathbf{C} \in \widetilde{\mathcal{C}}, \boldsymbol{\xi} \in \Delta_d} \max_{\boldsymbol{\lambda} \in \mathbb{R}^d, \boldsymbol{\mu} \in \mathbb{R}^K_+} \mathcal{L}^{\text{con}}(\mathbf{C}, \xi, \boldsymbol{\lambda}, \boldsymbol{\mu}) \\
&= \min_{\mathbf{C} \in \widetilde{\mathcal{C}}, \boldsymbol{\phi}(\mathbf{C}) \leq 0} \psi(\mathbf{C}). \qquad (33)
\end{aligned}$$

Recall that Algorithm 7 is designed to find the minimum of ψ over $\mathcal{C}$ (subject to constraints $\boldsymbol{\phi}$). However, the exact same sequence of iterates would also apply for minimizing ove $\widetilde{\mathcal{C}}$, and hence the sequence of iterates $\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t$ also approximately maximise $\widehat{f}^{\text{con}}$. Then by Theorem 44 and Equation (32) we have,

$$\begin{aligned}
\max_{\boldsymbol{\lambda} \in \mathbb{R}^d, \boldsymbol{\mu} \in \mathbb{R}^K_+} \widehat{\widetilde{f}}^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) &\leq \max_{0 \leq t \leq T} \widehat{\widetilde{f}}^{\text{con}}(\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) + \left(4a\sqrt{d+K}\right) \cdot \exp\left(\frac{-T}{2(d+K)^2}\right) + \tau \\
&\leq \max_{0 \leq t \leq T} \widehat{f}^{\text{con}}(\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t) + a\rho + \left(4a\sqrt{d+K}\right) \cdot \exp\left(\frac{-T}{2(d+K)^2}\right) + \tau \\
&\leq \max_{\boldsymbol{\lambda} \in \mathbb{R}^d, \boldsymbol{\mu} \in \mathbb{R}^K_+} \widehat{f}^{\text{con}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) + \left(4a\sqrt{d+K}\right) \cdot \exp\left(\frac{-T}{2(d+K)^2}\right) + 2\tau
\end{aligned}$$

Putting the above together with Equation (33) we get,

$$\min_{\mathbf{C} \in \widetilde{\mathcal{C}}, \boldsymbol{\phi}(\mathbf{C}) \leq 0} \psi(\mathbf{C}) \leq \min_{\mathbf{C} \in \mathcal{C}, \boldsymbol{\phi}(\mathbf{C}) \leq 0} \psi(\mathbf{C}) + \left(4a\sqrt{d+K}\right) \cdot \exp\left(\frac{-T}{2(d+K)^2}\right) + 2\tau$$

where $\tau = a(\rho + 2\sqrt{d}\rho')$. which completes the proof. □

Lemma 46. *Let* $\boldsymbol{\alpha}^* \in \operatorname*{argmin}_{\boldsymbol{\alpha}\in\Delta_T, \boldsymbol{\phi}(\sum_t \alpha_t \mathbf{C}^t)\leq 0} \psi\left(\sum_{i=0}^{T-1} \alpha_i \mathbf{C}^i\right)$. *Then:*

$$\psi\left(\sum_{i=0}^{T-1} \alpha_i^* \mathbf{C}[h^i]\right) \leq \min_{\mathbf{C}\in\widetilde{\mathcal{C}}, \boldsymbol{\phi}(\mathbf{C})\leq 0} \psi(\mathbf{C}) + 2\tau;$$
$$\phi_k\left(\sum_{i=0}^{T-1} \alpha_i^* \mathbf{C}[h^i]\right) \leq \tau,$$

where $\widetilde{\mathbf{C}} = \text{conv}(\{\mathbf{C}[h^0], \ldots, \mathbf{C}[h^{T-1}]\})$.

Proof. Let $\boldsymbol{\beta}^* \in \operatorname*{argmin}_{\boldsymbol{\beta}\in\Delta_T, \boldsymbol{\phi}(\sum_t \beta_t \mathbf{C}[h^t])\leq 0} \psi\left(\sum_{i=0}^{T-1} \beta_i \mathbf{C}[h^i]\right)$ denote the coefficients obtained by solving a similar minimization problem with the estimates $\mathbf{C}^t$ replaced with the true confusion matrices $C[h^t]$. First, we note that $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$ exist because h_0 (and in turn, $\mathbf{C}[h^0] = \mathbf{C}^0$) is strictly feasible.

$$\begin{aligned}
\psi\left(\sum_{i=0}^{T-1} \alpha_i^* \mathbf{C}[h^i]\right) &= \psi\left(\sum_{i=0}^{T-1} \alpha_i^* \mathbf{C}^i + \sum_{i=0}^{T-1} \alpha_i^* (\mathbf{C}[h^i] - \mathbf{C}^i)\right) \\
&\leq \psi\left(\sum_{i=0}^{T-1} \alpha_i^* \mathbf{C}^i\right) + L\rho'\sqrt{d} \\
&= \min_{\boldsymbol{\alpha}\in\Delta_T} \psi\left(\sum_{i=0}^{T-1} \alpha_i \mathbf{C}^i\right) + L\rho'\sqrt{d} \\
&\leq \psi\left(\sum_{i=0}^{T-1} \beta_i^* \mathbf{C}^i\right) + L\rho'\sqrt{d} \\
&= \psi\left(\sum_{i=0}^{T-1} \beta_i^* \mathbf{C}[h^i] + \sum_{i=0}^{T-1} \beta_i^* (\mathbf{C}^i - \mathbf{C}[h^i])\right) + L\rho'\sqrt{d} \\
&\leq \psi\left(\sum_{i=0}^{T-1} \beta_i^* \mathbf{C}[h^i]\right) + 2L\rho'\sqrt{d} \\
&= \min_{\boldsymbol{\beta}\in\Delta_T, \boldsymbol{\phi}(\sum_t \beta_t \mathbf{C}[h^t])\leq 0} \psi\left(\sum_{i=0}^{T-1} \beta_i \mathbf{C}[h^i]\right) + 2L\rho'\sqrt{d} \\
&= \min_{\mathbf{C}\in\widetilde{\mathcal{C}}, \boldsymbol{\phi}(\mathbf{C})\leq 0} \psi(\mathbf{C}) + 2L\rho'\sqrt{d} \\
&\leq \min_{\mathbf{C}\in\widetilde{\mathcal{C}}, \boldsymbol{\phi}(\mathbf{C})\leq 0} \psi(\mathbf{C}) + 2\tau,
\end{aligned}$$

where the first and third inequality above are due to the Lipschitzness of ψ.

Using a similar argument as above, we get for all $k \in [K]$,

$$\phi_k\left(\sum_{i=0}^{T-1} \alpha_i^* \mathbf{C}[h^i]\right) = \phi_k\left(\sum_{i=0}^{T-1} \alpha_i^* \mathbf{C}^i + \sum_{i=0}^{T-1} \alpha_i^* (\mathbf{C}[h^i] - \mathbf{C}^i)\right)$$

$$\leq \phi_k\left(\sum_{i=0}^{T-1}\alpha_i^*\mathbf{C}^i\right) + L\rho'\sqrt{d}$$
$$\leq 0 + L\rho'\sqrt{d} \leq \tau,$$

where the first inequality above is due to the Lipschitzness of ϕ, and the second inequality is due to the property of $\boldsymbol{\alpha}^*$ being chosen from a set such that the weighted combination of $\mathbf{C}^i$ is feasible.

We are now ready to prove Theorem 18. □

Proof of Theorem 18. Let $\boldsymbol{\alpha}^* \in \operatorname*{argmin}_{\boldsymbol{\alpha}\in\Delta_T, \boldsymbol{\phi}(\sum_t \alpha_t \mathbf{C}^t)\leq \mathbf{0}} \psi\left(\sum_{i=0}^{T-1}\alpha_i\mathbf{C}^i\right)$. Let $\bar{d} = d + K$. Putting Lemmas 45 and 46 together we get,

$$\psi(\mathbf{C}[\bar{h}]) = \psi\left(\sum_{i=0}^{T-1}\alpha_i^*\mathbf{C}[h^i]\right)$$
$$\leq \min_{\mathbf{C}\in\mathcal{C}, \boldsymbol{\phi}(\mathbf{C})\leq 0} \psi(\mathbf{C}) + \left(4a\sqrt{\bar{d}}\right)\cdot\exp\left(\frac{-T}{2(\bar{d})^2}\right) + 4\tau$$
$$= \min_{\mathbf{C}\in\mathcal{C}, \boldsymbol{\phi}(\mathbf{C})\leq 0} \psi(\mathbf{C}) + \left(4a\sqrt{\bar{d}}\right)\cdot\exp\left(\frac{-T}{2(\bar{d})^2}\right) + 4\tau$$

We now set $T = 2\bar{d}^2\log\left(\frac{\bar{d}}{\epsilon}\right)$ to obtain

$$\psi(\mathbf{C}[\bar{h}]) \leq \min_{\mathbf{C}\in\mathcal{C}, \boldsymbol{\phi}(\mathbf{C})\leq 0} \psi(\mathbf{C}) + \left(4a\right)\epsilon + 4\tau$$

The feasibility inequality then follows easily from Lemma 46

$$\phi_k(\mathbf{C}[\bar{h}]) = \phi_k\left(\sum_{i=0}^{T-1}\alpha_i^*\mathbf{C}[h^i]\right) \quad \leq \tau.$$

for all $k \in [K]$. □

A.11 Proof of Theorem 19 (Bisection for Constrained Problems)

Theorem ((Restated) Convergence of ConBisection algorithm)**.** *Fix $\epsilon \in (0,1)$. Let $\psi : [0,1]^d \to [0,1]$ be such that $\psi(\mathbf{C}) = \frac{\langle \mathbf{A},\mathbf{C}\rangle}{\langle \mathbf{B},\mathbf{C}\rangle}$, where $\mathbf{A},\mathbf{B} \in [0,1]^d$, and $\min_{\mathbf{C}\in\mathcal{C}}\langle \mathbf{B},\mathbf{C}\rangle = b$ for some $b > 0$. Let $\phi_1,\ldots,\phi_K : [0,1]^d \to [-1,1]$ be convex and L-Lipschitz w.r.t. the ℓ_2-norm. Let Ω in Algorithm 8 be a (ρ,ρ',δ)-approximate LMO for sample size N. Suppose the strict feasibility condition in Assumption 1 holds for radius $r > 0$. Let Λ, Ξ, ω and ω' in the call to Algorithm 6 be set as in Theorem 17 with Lipschitz constant $L' = \max\{L, \|\mathbf{A}\|_2 + \|\mathbf{B}\|_2\}$. Let $\bar{h}$ be a classifier returned by Algorithm 8 when run for T outer iterations and T' inner iterations. Then with probability $\geq 1-\delta$ over draw of $S \sim D^N$, after $T = \log(1/\epsilon)$ outer iterations and $T' = \mathcal{O}(K/\epsilon^2)$ inner iterations:*

$$\textbf{\textit{Optimality}}: \quad \psi(\mathbf{C}[\bar{h}]) \leq \min_{\mathbf{C}\in\mathcal{C}:\, \boldsymbol{\phi}(\mathbf{C})\leq \mathbf{0}} \psi(\mathbf{C}) + \mathcal{O}\left(\kappa(\epsilon + \rho^{\text{eff}})\right);$$

$$\textbf{\textit{Feasibility}}: \quad \phi_k(\mathbf{C}[\bar{h}]) \leq \mathcal{O}\left(L'(\epsilon + \rho^{\text{eff}})\right), \ \forall k \in [K],$$

where $\kappa = L'/b$ and $\rho^{\text{eff}} = \rho + \sqrt{d}\rho'$.

The proof follows similar steps as that for Theorem 15. We will first state a couple of lemmas:

Lemma 47 (Invariant in Algorithm 8)**.** *Under the assumptions made in Theorem 19, the following invariant is true at the end of each iteration $0 \leq t \leq T$ of Algorithm 8:*

$$\min_{\mathbf{C} \in \mathcal{C}:\, \phi_k(\mathbf{C}) \leq 0, \forall k} \psi(\mathbf{C}) \geq \alpha^t \, - \, \mathcal{O}\left(\kappa(\epsilon + \rho^{\text{eff}})\right);$$
$$\psi(\mathbf{C}[h^t]) \, < \, \beta^t \, + \, \mathcal{O}\left(\kappa(\epsilon + \rho^{\text{eff}})\right);$$
$$\phi_k(\mathbf{C}[h^t]) \, \leq \, \mathcal{O}\left(L'(\epsilon + \rho^{\text{eff}})\right), \, \forall k \in [K].$$

where L', κ, and ρ^{eff} are defined as in Theorem 19.

Proof. We shall prove this lemma by mathematical induction on the iteration number t. For $t = 0$, the invariant holds trivially as $0 \leq \psi(\mathbf{C}[h^0]) \leq 1$ and h^0 satisfies the constraints. Assume the invariant holds at the end of iteration $t-1 \in \{0, \ldots, T-1\}$; we shall prove that the invariant holds at the end of iteration t.

First note that the linear function $\psi'(\mathbf{C}) = \langle \mathbf{A} - \gamma^t \mathbf{B}, \, \mathbf{C} \rangle$ in step 6 of the algorithm is Lipschitz w.r.t. the ℓ_2-norm with Lipschitz parameter of at most $\|\mathbf{A} - \gamma^t \mathbf{B}\|_2 \leq \|\mathbf{A}\|_2 + \|\mathbf{B}\|_2 \leq L'$. We then have from Theorem 17 that the classifier g^t returned by the ConGDA algorithm (Algorithm 6) after $T' = \mathcal{O}(K/\epsilon^2)$ runs enjoys the following guarantee:

$$\langle \mathbf{A} - \gamma^t \mathbf{B}, \, \mathbf{C}[g^t] \rangle \rangle \;\; \leq \;\; \min_{\mathbf{C} \in \mathcal{C}:\, \phi_k(\mathbf{C}) \leq 0, \forall k} \langle \mathbf{A} - \gamma^t \mathbf{B}, \, \mathbf{C} \rangle \, + \, \mathcal{O}\left(L'(\epsilon + \rho^{\text{eff}})\right); \tag{34}$$
$$\phi_k(\mathbf{C}[g^t]) \;\; \leq \;\; \mathcal{O}\left(L'(\epsilon + \rho^{\text{eff}})\right), \, \forall k \in [K], \tag{35}$$

where we have used the fact that both $\psi'(\mathbf{C}) = \langle \mathbf{A} - \gamma^t \mathbf{B}, \, \mathbf{C} \rangle$ and $\phi_k(\mathbf{C})$s are L'-Lipschitz w.r.t. the ℓ_2-norm. We further have that from the property of the LMO (Definition 11) used in turn by Algorithm 5 that:

$$\|\mathbf{C}^t - \mathbf{C}[g^t]\|_\infty \leq \rho' \tag{36}$$

We now consider two cases at iteration t. In the first case, $\psi(\mathbf{C}^t) < \gamma^t$, leading to the assignments $\alpha^t = \alpha^{t-1}$, $\beta^t = \gamma^t$, and $h^t = g^t$. We then have:

$$\begin{aligned}
\langle \mathbf{A} - \gamma^t \mathbf{B}, \, \mathbf{C}[g^t] \rangle \rangle \;\; &= \;\; \langle \mathbf{A} - \gamma^t \mathbf{B}, \, \mathbf{C}^t \rangle \, + \, \langle \mathbf{A} - \gamma^t \mathbf{B}, \mathbf{C}[g^t] - \mathbf{C}^t \rangle \\
&\leq \;\; \langle \mathbf{A} - \gamma^t \mathbf{B}, \, \mathbf{C}^t \rangle \, + \, \|\mathbf{A} - \gamma^t \mathbf{B}\|_1 \rho' \\
&= \;\; \langle \mathbf{B}, \, \mathbf{C}^t \rangle \big(\psi(\mathbf{C}^t) \, - \, \gamma^t\big) \, + \, \|\mathbf{A} - \gamma^t \mathbf{B}\|_1 \rho' \\
&\leq \;\; 0 \, + \, \|\mathbf{A} - \gamma^t \mathbf{B}\|_1 \rho' \\
&\leq \;\; (\|\mathbf{A}\|_1 + \|\mathbf{B}\|_1) \rho' \\
&\leq \;\; (\|\mathbf{A}\|_2 + \|\mathbf{B}\|_2) \sqrt{d} \rho' \, \leq \, L' \sqrt{d} \rho' \, < \, L'(\epsilon + \rho^{\text{eff}}),
\end{aligned}$$

where the second step follows from Hölder's inequality and (36), the fourth step follows from our case assumption that $\psi(\mathbf{C}^t) \leq \gamma^t$ and from $\langle \mathbf{B}, \, \mathbf{C}^t \rangle > 0$, the fifth step follows from the triangle inequality and $0 \leq \gamma^t \leq 1$, and the sixth step uses the fact that $\|\mathbf{z}\|_1 \leq \sqrt{d} \|\mathbf{z}\|_2$. The above inequality, along with the fact that $\langle \mathbf{B}, \mathbf{C} \rangle \geq b$, further gives us:

$$\frac{\langle \mathbf{A}, \, \mathbf{C}[g^t] \rangle}{\langle \mathbf{B}, \, \mathbf{C}[g^t] \rangle} \, < \, \gamma^t \, + \, \frac{L'}{b}(\epsilon + \rho^{\text{eff}}) \, = \, \beta^t \, + \, \kappa(\epsilon + \rho^{\text{eff}}).$$

In other words,

$$\psi\big(\mathbf{C}[h^t]\big) \;=\; \psi(\mathbf{C}[g^t]) \;=\; \frac{\langle \mathbf{A},\, \mathbf{C}^D[g^t]\rangle}{\langle \mathbf{B},\, \mathbf{C}^D[g^t]\rangle} \;<\; \beta^t \;+\; \mathcal{O}(\kappa(\epsilon + \rho^{\text{eff}})).$$

Moreover, by our assumption that the invariant holds at the end of iteration $t-1$, we have

$$\min_{\mathbf{C}\in\mathcal{C}:\, \phi_k(\mathbf{C})\leq 0, \forall k} \psi(\mathbf{C}) \geq \alpha^{t-1} \;-\; \mathcal{O}\left(\kappa(\epsilon + \rho^{\text{eff}})\right) = \alpha^t \;-\; \mathcal{O}\left(\kappa(\epsilon + \rho^{\text{eff}})\right).$$

Further, from (35), $\phi_k(\mathbf{C}[h^t]) = \phi_k(\mathbf{C}[g^t]) \leq \mathcal{O}(L'\bar{\rho}), \forall k$. Thus under the first case, the invariant holds at the end of iteration t.

In the second case, $\psi(\mathbf{C}^t) \geq \gamma^t$ at iteration t, which would lead to the assignments $\alpha^t = \gamma^t$, $\beta^t = \beta^{t-1}$, and $h^t = h^{t-1}$. Since the invariant is assumed to hold at the end of iteration $t-1$, we have

$$\psi(\mathbf{C}[h^t]) \;=\; \psi(\mathbf{C}[h^{t-1}]) \;\leq\; \beta^{t-1} \;+\; \mathcal{O}(\kappa(\epsilon + \rho^{\text{eff}})) \;=\; \beta^t \;+\; \mathcal{O}(\kappa(\epsilon + \rho^{\text{eff}})). \tag{37}$$

Next for $\mathbf{C}^* \in \underset{\mathbf{C}\in\mathcal{C}:\, \boldsymbol{\phi}(\mathbf{C})\leq \mathbf{0}}{\operatorname{argmin}} \langle \mathbf{A} - \gamma^t\mathbf{B},\, \mathbf{C}\rangle$, we have from (34),

$$\begin{aligned}
\langle \mathbf{A} - \gamma^t\mathbf{B},\, \mathbf{C}^*\rangle \;&\geq\; \langle \mathbf{A} - \gamma^t\mathbf{B},\, \mathbf{C}[h^t]\rangle \;-\; \mathcal{O}(L'(\epsilon + \rho^{\text{eff}})) \\
&\geq\; \langle \mathbf{A} - \gamma^t\mathbf{B},\, \mathbf{C}^t\rangle \;-\; \|\mathbf{A} - \gamma^t\mathbf{B}\|_1 \|\mathbf{C}^t - \mathbf{C}[h^t]\|_\infty \;-\; \mathcal{O}(L'(\epsilon + \rho^{\text{eff}})) \\
&\geq\; \langle \mathbf{A} - \gamma^t\mathbf{B},\, \mathbf{C}^t\rangle \;-\; \|\mathbf{A} - \gamma^t\mathbf{B}\|_1 \rho' \;-\; \mathcal{O}(L'(\epsilon + \rho^{\text{eff}})) \\
&=\; \langle \mathbf{B},\, \mathbf{C}^t\rangle(\psi(\mathbf{C}^t) - \gamma^t) \;-\; \|\mathbf{A} - \gamma^t\mathbf{B}\|_1 \rho' \;-\; \mathcal{O}(L'(\epsilon + \rho^{\text{eff}})) \\
&\geq\; \langle \mathbf{B},\, \mathbf{C}^t\rangle(0) \;-\; \|\mathbf{A} - \gamma^t\mathbf{B}\|_1 \rho' \;-\; \mathcal{O}(L'(\epsilon + \rho^{\text{eff}})) \\
&\geq\; -(\|\mathbf{A}\|_2 + \|\mathbf{B}\|_2)\sqrt{d}\rho' \;-\; \mathcal{O}(L'(\epsilon + \rho^{\text{eff}})) \;=\; -\mathcal{O}(L'(\epsilon + \rho^{\text{eff}})),
\end{aligned}$$

where the first step follows from the property of the LMO, the second step follows from Hölder's inequality, the third step uses (36), the fifth step follows from our case assumption that $\psi(\mathbf{C}^t) \geq \gamma^t$ and $\langle \mathbf{B},\, \mathbf{C}^t\rangle > 0$, the last step follows from the triangle inequality, $0 \leq \gamma^t \leq 1$, and the fact that $\|\mathbf{z}\|_1 \geq \|\mathbf{z}\|_2$. In particular, we have for all $\mathbf{C} \in \mathcal{C}$ such that $\phi_k(\mathbf{C}) \leq 0, \forall k$,

$$\langle \mathbf{A} - \gamma^t\mathbf{B},\, \mathbf{C}\rangle \;\geq\; -\mathcal{O}(L'(\epsilon + \rho^{\text{eff}})),$$

or

$$\frac{\langle \mathbf{A},\, \mathbf{C}\rangle}{\langle \mathbf{B},\, \mathbf{C}\rangle} \;\geq\; \gamma^t - \mathcal{O}\left(\frac{L'}{\langle \mathbf{B},\, \mathbf{C}\rangle}(\epsilon + \rho^{\text{eff}})\right) \;\geq\; \gamma^t - \mathcal{O}\left(\frac{L'}{b}(\epsilon + \rho^{\text{eff}})\right) \;=\; \gamma^t - \mathcal{O}\left(\kappa(\epsilon + \rho^{\text{eff}})\right).$$

In other words,

$$\min_{\mathbf{C}\in\mathcal{C}:\, \phi_k(\mathbf{C})\leq 0, \forall k} \psi(\mathbf{C}) \;\geq\; \gamma^t \;-\; \mathcal{O}\left(\kappa(\epsilon + \rho^{\text{eff}})\right) \;=\; \alpha^t - \mathcal{O}\left(\kappa(\epsilon + \rho^{\text{eff}})\right).$$

By combining the above with (37) and noting that $\phi_k(\mathbf{C}[h^t]) = \phi_k(\mathbf{C}[h^{t-1}]) \leq \mathcal{O}\left(L'(\epsilon + \rho^{\text{eff}})\right)$, $\forall k$, we can see that the invariant holds in iteration t under this case as well. This completes the proof of the lemma. □

Lemma 48 (Multiplicative progress in each iteration of Algorithm 8)**.** *Let ψ be as defined in Theorem 19. Then the following is true in each iteration $1 \leq t \leq T$ of Algorithm 8:*

$$\beta^t - \alpha^t \;=\; \frac{1}{2}\big(\beta^{t-1} - \alpha^{t-1}\big).$$

Proof. We consider two cases in each iteration of Algorithm 8. If in an iteration $t \in \{1, \ldots, T\}$, $\psi(\mathbf{C}^t) < \gamma^t$, leading to the assignment $\beta^t = \gamma^t$, then

$$\beta^t - \alpha^t \;=\; \gamma^t - \alpha^{t-1} \;=\; \frac{\alpha^{t-1} + \beta^{t-1}}{2} - \alpha^{t-1} \;=\; \frac{1}{2}(\beta^{t-1} - \alpha^{t-1}).$$

On the other hand, if $\psi(\mathbf{C}^t) \geq \gamma^t$, leading to the assignment $\alpha^t = \gamma^t$, then

$$\beta^t - \alpha^t \;=\; \beta^{t-1} - \gamma^t \;=\; \beta^{t-1} - \frac{\alpha^{t-1} + \beta^{t-1}}{2} \;=\; \frac{1}{2}\big(\beta^{t-1} - \alpha^{t-1}\big).$$

Thus in both cases, the statement of the lemma is seen to hold. □

We are now ready to prove Theorem 19.

Proof of Theorem 19. For the classifier $\bar{h} = h^T$ output by Algorithm 8 after T iterations, we have from Lemma 47,

$$\begin{aligned}
\psi\big(\mathbf{C}[h^T]\big) \;-\; \min_{\mathbf{C} \in \mathcal{C}:\, \phi_k(\mathbf{C}) \leq 0, \forall k} \psi(\mathbf{C}) \;&<\; \beta^T - \alpha^T \;+\; \mathcal{O}(\kappa(\epsilon + \rho^{\text{eff}})) \\
&\leq\; 2^{-T}\big(\beta^0 - \alpha^0\big) \;+\; \mathcal{O}(\kappa(\epsilon + \rho^{\text{eff}})) \\
&=\; 2^{-T}\big(1 - 0\big) \;+\; \mathcal{O}(\kappa(\epsilon + \rho^{\text{eff}})) \\
&=\; 2^{-T} \;+\; \mathcal{O}(\kappa(\epsilon + \rho^{\text{eff}})),
\end{aligned}$$

where the second step follows from repeated application of Lemma 48. Additionally, we have from Lemma 47, $\phi_k(\mathbf{C}[\bar{h}]) \;\leq\; \mathcal{O}\left(L'(\epsilon + \rho^{\text{eff}})\right),\ \forall k \in [K]$. Setting $T = \log(1/\epsilon)$ completes the proof. □

A.12 Proof of Theorem 20

Theorem ((Restated) Regret bound for plug-in LMO)**.** *Fix $\delta \in (0, 1)$. Then with probability $\geq 1 - \delta$ over draw of sample $S \sim D^N$, for any loss matrix $\mathbf{L} \in \mathbb{R}^d_+$ with $\|\mathbf{L}\|_\infty = 1$, the classifier and confusion matrix $(\widehat{g}, \widehat{\mathbf{\Gamma}})$ returned by Algorithm 9 satisfies:*

$$\langle \mathbf{L}, \mathbf{C}[\widehat{g}] \rangle \;\leq\; \min_{h: \mathcal{X} \to \Delta_n} \langle \mathbf{L}, \mathbf{C}[h] \rangle + \mathbf{E}_X\big[\big\|\widehat{\boldsymbol{\eta}}(X) \;-\; \boldsymbol{\eta}(X)\big\|_1\big];$$

$$\|\mathbf{C}[\widehat{g}] \;-\; \widehat{\mathbf{\Gamma}}\|_\infty \;\leq\; \mathcal{O}\left(\sqrt{\frac{d \log(n) \log(N) + \log(d/\delta)}{N}}\right).$$

Proof. For simplicity, we will represent both $\mathbf{L}$ and $\mathbf{C}$ as $n \times n$ matrices instead of flattened n^2-dimensional vectors. Let us denote the columns of $\mathbf{L}$ by $\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_n$, where $\ell_j = [L_{1,j}, L_{2,j}, \ldots, L_{n,j}]^\top$.

We can then re-write:

$$\begin{aligned}\langle \mathbf{L}, \mathbf{C}[h] \rangle = \sum_{i,j} L_{ij}\, C_{ij}[h] \;&=\; \sum_{i,j} \mathbf{E}_X \left[\eta_i(X)\, L_{ij}\, \mathbf{1}(h(X) = j) \right] \\ &= \sum_{j=1}^{n} \mathbf{E}_X \left[\mathbf{1}(h(X) = j)\, \boldsymbol{\eta}(X)^\top \boldsymbol{\ell}_j \right] \;=\; \mathbf{E}_X \left[\boldsymbol{\eta}(X)^\top \boldsymbol{\ell}_{h(X)} \right].\end{aligned}$$

Let h^* be the Bayes optimal classifier for the linear metric $\langle \mathbf{L}, \mathbf{C}[\widehat{h}] \rangle$. For the first part, we bound the $\mathbf{L}$-regret as follows:

$$\begin{aligned}&\langle \mathbf{L}, \mathbf{C}[\widehat{g}] \rangle - \langle \mathbf{L}, \mathbf{C}[h^*] \rangle \\ &= \mathbf{E}_X \left[\boldsymbol{\eta}(X)^\top \boldsymbol{\ell}_{\widehat{g}(X)} \right] - \mathbf{E}_X \left[\boldsymbol{\eta}(X)^\top \boldsymbol{\ell}_{h^*(X)} \right] \\ &= \mathbf{E}_X \left[\widehat{\boldsymbol{\eta}}(X)^\top \boldsymbol{\ell}_{\widehat{g}(X)} \right] + \mathbf{E}_X \left[(\boldsymbol{\eta}(X) - \widehat{\boldsymbol{\eta}}(X)^\top \boldsymbol{\ell}_{\widehat{g}(X)} \right] - \mathbf{E}_X \left[\boldsymbol{\eta}(X)^\top \boldsymbol{\ell}_{h^*(X)} \right] \\ &\leq \mathbf{E}_X \left[\widehat{\boldsymbol{\eta}}(X)^\top \boldsymbol{\ell}_{h^*(X)} \right] + \mathbf{E}_X \left[(\boldsymbol{\eta}(X) - \widehat{\boldsymbol{\eta}}(X)^\top \boldsymbol{\ell}_{\widehat{g}(X)} \right] - \mathbf{E}_X \left[\boldsymbol{\eta}(X)^\top \boldsymbol{\ell}_{h^*(X)} \right] \\ &= \mathbf{E}_X \left[(\boldsymbol{\eta}(X) - \widehat{\boldsymbol{\eta}}(X))^\top (\boldsymbol{\ell}_{\widehat{g}(X)} - \boldsymbol{\ell}_{h^*(X)}) \right] \\ &\leq \mathbf{E}_X \left[\left\| \boldsymbol{\eta}(X) - \widehat{\boldsymbol{\eta}}(X) \right\|_1 \cdot \left\| \boldsymbol{\ell}_{\widehat{g}(X)} - \boldsymbol{\ell}_{h^*(X)} \right\|_\infty \right] \\ &\leq \mathbf{E}_X \left[\left\| \boldsymbol{\eta}(X) - \widehat{\boldsymbol{\eta}}(X) \right\|_1 \right],\end{aligned}$$

where in the third step, we use the fact that $\widehat{g}(x) \;=\; \operatorname{argmin}^*_{j \in [n]} \widehat{\boldsymbol{\eta}}(x)^\top \boldsymbol{\ell}_j$; in the last step, we have use the fact that $\|\mathbf{L}\|_\infty = 1$.

For the second part, we denote the class of all plug-in classifiers constructed from a fixed class-probability estimator $\widehat{\boldsymbol{\eta}}$ by:

$$\mathcal{H} = \left\{ h : \mathcal{X} \to [n],\, h(x) \;=\; \operatorname{argmin}^*_{y \in [n]} \boldsymbol{\ell}_y^\top \widehat{\boldsymbol{\eta}}(x) \,|\, \mathbf{L} \in [0,1]^{n \times n} \right\},$$

and provide a uniform convergence bound over all classifiers in $\mathcal{H}$, and in turn applies to the classifier $\widehat{g}$ output by Algorithm 9.

For any $a, b \in [n]$, we have

$$\begin{aligned}\sup_{h \in \mathcal{H}_{\mathbf{q}}} \left| \widehat{C}^S_{a,b}[h] - C_{a,b}[h] \right| \;&=\; \sup_{h \in \mathcal{H}} \left| \frac{1}{m} \sum_{i=1}^{m} \left(\mathbf{1}(y_i = a, h(x_i) = b) - \mathbf{E}[\mathbf{1}(Y = a, h(X) = b)] \right) \right| \\ &=\; \sup_{h \in \mathcal{H}^b} \left| \frac{1}{m} \sum_{i=1}^{m} \left(\mathbf{1}(y_i = a, h(x_i) = 1) - \mathbf{E}[\mathbf{1}(Y = a, h(X) = 1)] \right) \right|,\end{aligned}$$

where for a fixed $b \in [n]$,

$$\mathcal{H}^b \;=\; \left\{ h : \mathcal{X} \to \{0,1\} : \exists \mathbf{L} \in [0,1]^{n \times n}, \forall x \in \mathcal{X},\, h(x) = \mathbf{1} \left(b = \operatorname{argmin}^*_{y \in [n]} \boldsymbol{\ell}_y^\top \widehat{\boldsymbol{\eta}}(x) \right) \right\}.$$

The set $\mathcal{H}^b$ can be seen as hypothesis class whose concepts are the intersection of n halfspaces in $\mathbb{R}^n$ (corresponding to $\widehat{\boldsymbol{\eta}}(x)$) through the origin. Hence we have from Lemma 3.2.3 of Blumer et al. (1989) that the VC-dimension of $\mathcal{H}^b$ is at most $2n^2 \log(3n)$.

From standard uniform convergence arguments we have that for each $a, b \in [n]$, the following holds with at least probability $1 - \delta$ (over draw of $S \sim D^N$),

$$\sup_{h \in \mathcal{H}} \left| \widehat{C}^S_{a,b}[h] - C_{a,b}[h] \right| \leq \mathcal{O}\left(\sqrt{\frac{n^2 \log(n) \log(N) + \log(\frac{1}{\delta})}{N}} \right).$$

Applying union bound over all $a, b \in [n]$, we have that the following holds with probability $\geq 1 - \delta$:

$$\left|\left| \widehat{\mathbf{C}}^S[\widehat{g}] - \mathbf{C}[\widehat{g}] \right|\right|_\infty \leq \sup_{h \in \mathcal{H}} \left|\left| \widehat{\mathbf{C}}^S[h] - \mathbf{C}[h] \right|\right|_\infty \leq \mathcal{O}\left(\sqrt{\frac{n^2 \log(n) \log(N) + \log(\frac{n^2}{\delta})}{N}} \right).$$

Plugging $d = n^2$ completes the proof. □

Appendix B. Additional Experimental Details

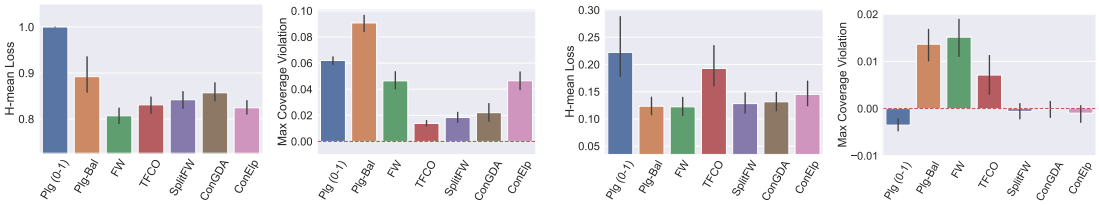
B.1 Hyper-parameter Selection

We run the Frank-Wolfe and GDA algorithms for 5000 LMO calls, and the ellipsoid algorithm for 1000 LMO calls. We run the constrained algorithms for 10000, 10000 and 1000 LMO calls respectively. The unconstrained Frank-Wolfe algorithm has no other hyper-parameters to tune. The GDA algorithm has two step-size parameters ω and ω', which we tune using a two-dimensional grid-search over $\{0.001, 0.01, 0.1\}^2$, picking the parameters that yield the lowest objective on the training set. For the ellipsoid algorithm, we fix the initial ellipsoid radius a to 1000.

The constrained counterpart to the Frank-Wolfe algorithm (SplitFW) in Algorithm 5 has two additional hyper-parameters: the weight on the quadratic penalty ζ, which we set to 10, and the step-size ω, for which, we adopt the same schedule used by Gidel et al. (2018), and set it to 0.5 for first $T/3$ iterations, 0.1 for the next $T/3$ iterations, and 0.001 for the final $T/3$ iterations. Additionally, we find it sufficient to avoid the explicit line search for γ^t in line 7 and instead set to $\frac{2}{t+2}$, akin to the standard Frank-Wolfe setup. For the constrained version of GDA algorithm, we set the step-sizes $\omega_{\boldsymbol{\lambda}} = \omega_{\boldsymbol{\mu}} = \omega'$, and tune $\omega_{\boldsymbol{\xi}}$ and ω' using the same the two-dimensional grid search used for unconstrained GDA, picking among those that satisfy the constraints on the training set, the ones with the least training objective (When none of the parameters satisfy the constraints, we pick the one with the minimum constraint violation). The hyper-parameters for the constrained ellipsoid algorithm were chosen in the same way as the unconstrained version. For TFCO, we tuned the learning rates for the model and constraint from $\{0.001, 0.01, 0.1\}$ and ran it for 5000 iterations.

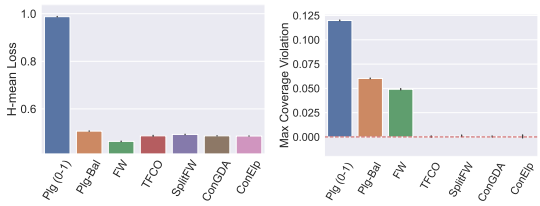
B.2 Additional Details for CIFAR Case Studies

Below, we list the five super-classes in the CIFAR-55 dataset described in Section 8.8, and the 10 classes that each of them comprise of: (i) **Flowers and Fruits:** Orchid, Poppy, Rose, Sunflower, Tulip, Mushroom, Orange, Pear, Apples, and Sweet Pepper. (ii) **Aquatic Animals:** Beaver, Dolphin, Otter, Aquarium Fish, Ray, Flat Fish, Shark, Trout, Whale, and Seal. (iii) **Household Items:** Clock, Bed, Chair, Couch, Keyboard, Telephone, Television, Wardrobe, Table, and Lamp. (iv) **Large Outdoor Scenes:** Bridge, Castle, House, Road, Mountain, Skyscraper, Cloud, Forest, Plain, and Sea. (v) **Mammals:** Camel, Cattle, Chimpanzee, Elephant, Kangaroo, Porcupine, Pos-



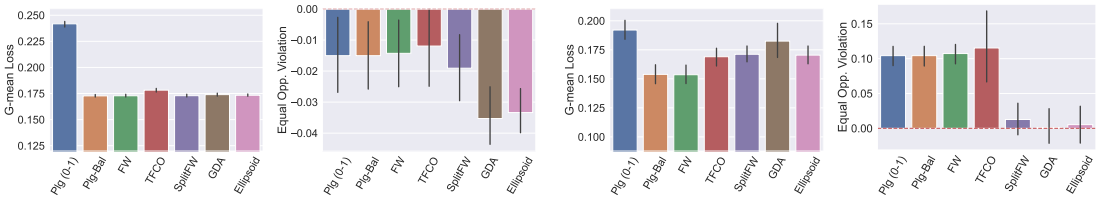
(a) Abalone

(b) MACHO



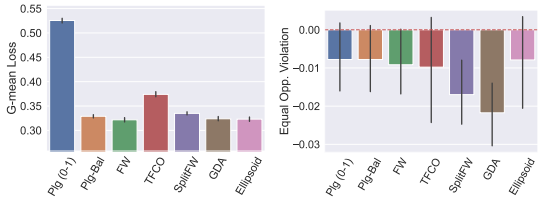
(c) CovType

Figure 13: Optimizing the H-mean loss subject to the coverage constraint $\max_i |\sum_j C_{ji} - \pi_i| \leq 0.01$. The plots on the left show the H-mean loss on the test set and those on the right show the coverage violation $\max_i |\sum_j C_{ji} - \pi_i| - 0.01$ on the test set. *Lower* H-mean value are *better*, and the constraint values need to be ≤ 0.



(a) Adult

(b) Crimes

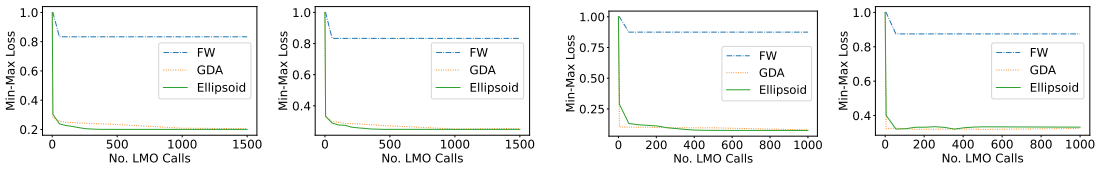


(c) Default

Figure 14: Optimizing the G-mean loss subject to the Equal Opportunity constraint ≤ 0.01. The plots on the left show the G-mean loss on the test set and those on the right show the equal opportunity violation (needs to be ≤ 0) on the test set. *Lower* G-mean values are *better*.

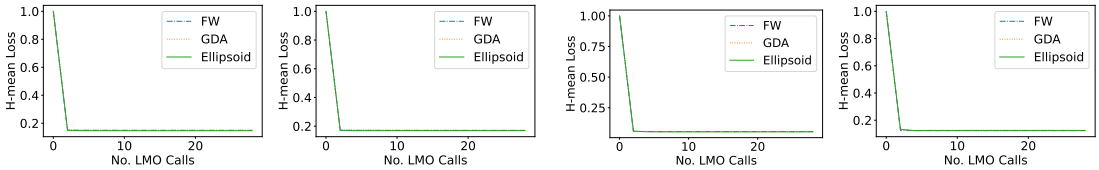
sum, Raccoon, Fox, and Skunk. We employ standard data augmentation techniques on the CIFAR datasets by applying random crops and horizontal flips.[††]

[††]The learning rate schedules were adopted from: `https://github.com/huyvnphan/PyTorch_CIFAR10`.



(a) SatImage (b) MACHO

Figure 15: Optimizing the *Min-max* loss: Comparison of performance of the Frank-Wolfe, GDA and ellipsoid methods as a function of the number of LMO calls. The plot on the left is for train data and on the right is for test data. *Lower* values are *better*. Because the min-max loss is non-smooth, Frank-Wolfe is seen to converge to a sub-optimal classifier.

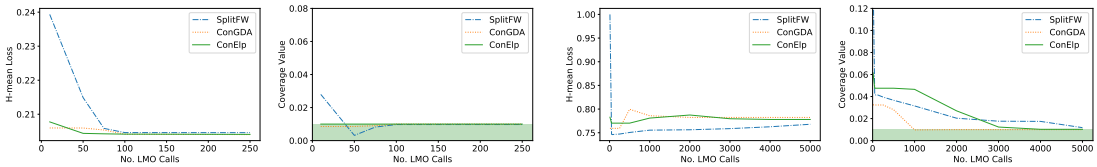


(a) SatImage (b) MACHO

Figure 16: Optimizing the *Hmean* loss: Comparison of performance of the Frank-Wolfe, GDA and ellipsoid methods as a function of the number of LMO calls. The plot on the left is for Train data and on the right is for Test data. *Lower* values are *better*.

B.3 Additional Experimental Results

We report the H-mean Loss and micro-F measures of a random classifier on our datasets in Table 13. We also present additional results for the experiments described in Section 8: (i) **Performance on Constrained Problems** (Section 8.5): See Figures 13– 14. (ii) **Practical Guidance on Algorithm Choice** (Section 8.6): See Figures 15–17. (iii) **Choice of LMO: Plug-in vs. Weighted Logistic Regression** (Section 8.7): See Table 12.



(a) Adult (b) Abalone

Figure 17: Optimizing the H-mean loss subject to the coverage constraint $\max_i |\sum_j C_{ji} - \pi_i| \leq 0.01$. The plots on the left show the H-mean loss on the train set and those on the right show the coverage violation $\max_i |\sum_j C_{ji} - \pi_i| - 0.01$ on the train set. *Lower* H-mean value are *better*, and the constraint values need to be ≤ 0.

Table 12: Comparison of the plug-in and weighted logistic regression (WLR) based LMOs on the task of optimizing the (convex) H-mean loss. The number of iterations, i.e. calls to the LMO, is fixed at *100*. *Lower* values are *better*. The results are averaged over 10 random train-test splits

Dataset	**FW**		**Ellipsoid**		**GDA**	
	Plugin	**WLR**	**Plugin**	**WLR**	**Plugin**	**WLR**
Aba	0.812 ± 0.017	$\mathbf{0.798 \pm 0.013}$	$\mathbf{0.815 \pm 0.017}$	0.817 ± 0.012	0.841 ± 0.032	$\mathbf{0.837 \pm 0.035}$
PgB	0.127 ± 0.039	$\mathbf{0.079 \pm 0.015}$	0.111 ± 0.026	$\mathbf{0.079 \pm 0.018}$	0.122 ± 0.032	$\mathbf{0.084 \pm 0.018}$
MAC	$\mathbf{0.124 \pm 0.017}$	0.245 ± 0.027	$\mathbf{0.125 \pm 0.017}$	0.247 ± 0.027	$\mathbf{0.124 \pm 0.016}$	0.206 ± 0.029
Sat	0.171 ± 0.007	$\mathbf{0.170 \pm 0.007}$	0.170 ± 0.006	$\mathbf{0.167 \pm 0.006}$	0.171 ± 0.007	$\mathbf{0.170 \pm 0.006}$
Cov	0.466 ± 0.001	$\mathbf{0.450 \pm 0.001}$	0.466 ± 0.001	$\mathbf{0.451 \pm 0.001}$	0.463 ± 0.001	$\mathbf{0.447 \pm 0.001}$

Table 13: Performance metrics of a Random Classifiers on the Dataset. *Lower Values are better.*

Dataset	H-mean Loss	micro-F1
Communities & Crime	0.506	0.503
COMPAS	0.501	0.504
Law School	0.501	0.499
Default	0.499	0.498
Adult	0.499	0.499
Abalone	0.940	0.918
Pgblk	0.839	0.794
MACHO	0.914	0.874
SatImage	0.835	0.832
CovType	0.858	0.857