Consistent Plug-in Classifiers for Complex Objectives and Constraints

Shiv Kumar Tavker
Indian Institute of Technology Madras, India
shivtavker@smail.iitm.ac.in

Harish G. Ramaswamy
Indian Institute of Technology Madras, India
hariguru@cse.iitm.ac.in

Harikrishna Narasimhan
Google Research, USA
hnarasimhan@google.com

Abstract

We present a consistent algorithm for constrained classification problems where the objective (e.g. F-measure, G-mean) and the constraints (e.g. demographic parity fairness, coverage) are defined by general functions of the confusion matrix. Our approach reduces the problem into a sequence of plug-in classifier learning tasks. The reduction is achieved by posing the learning problem as an optimization over the intersection of two sets: the set of confusion matrices that are achievable and those that are feasible. This decoupling of the constraint space then allows us to solve the problem by applying Frank-Wolfe style optimization over the individual sets. For objective and constraints that are convex functions of the confusion matrix, our algorithm requires $O(1/\epsilon^2)$ calls to the plug-in subroutine, which improves on the $O(1/\epsilon^3)$ calls needed by the reduction-based algorithm of Narasimhan (2018) [29]. We show empirically that our algorithm is competitive with prior methods, while being more robust to choices of hyper-parameters.

1 Introduction

In an increasing number of machine learning tasks, one is required to train a classifier with constraints on multiple metrics such as fairness, coverage, recall, etc [16, 17, 2, 9, 10]. Often, the objective and constraints in these problems are not simple metrics such as classification error, and may have a more complex non-decomposable structure, i.e. may not be expressible a simple average of errors on individual data points. Examples of such metrics include the F-measure and G-mean used in class-imbalanced problems [27, 24], the predictive parity criteria used in ML fairness [7], KL-divergence based metrics used in distribution matching tasks [12, 14], and many more.

A common feature of the above metrics is that they can all be defined as a function of a classifier's confusion matrix. We are therefore interested in constrained learning problems where the objectives and constraints are general functions of the confusion matrix. Our goal is to design a *statistically consistent* algorithm for solving these problems, i.e. an algorithm that converges in the limit of infinite training data to an optimal feasible classifier for these problems.

In previous work, Narasimhan (2018) [29] provide consistent algorithms for constrained learning problems by reducing them into a sequence of easy-to-solve sub-problems. Each of these sub-problems is a linear metric minimization task and involves learning a plug-in classifier, a classifier constructed by fine-tuning a threshold (or a weight matrix for multiclass problems) on a pre-trained class probability model. For convex functions of the confusion matrix, their method requires $O(1/\epsilon^3)$ calls to the plug-in learning routine to converge to a classifier that is ϵ-optimal and ϵ-feasible. In this

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

paper, we build on their work and provide an algorithm which requires only $O(1/\epsilon^2)$ calls to the plug-in routine to reach a classifier of the same quality.

Like the prior method, the key to our approach is to translate the constrained learning problem into an optimization problem over a finite dimensional space. While Narasimhan (2008) formulate this optimization problem over the space of confusion matrices that are achievable by a classifier, we formulate the problem over the intersection of *two* convex sets: the set of confusion matrices that are achievable, and the set of confusion matrices that are feasible, i.e. satisfy the constraints. The decoupling of the search space into two sets then allows us to adapt the Frank-Wolfe based algorithm of Gidel et al. (2018) [15] to solve the optimization. Our approach makes use of two oracle subroutines, both of which can be implemented efficiently. The first oracle minimizes a linear function over the space of achievable confusion matrices, which amounts to learning a plug-in classifier. The second performs a linear minimization over the space of feasible matrices, which is often a straight-forward convex program.

The proposed algorithm enjoys several practical benefits. Firstly, the algorithm is computationally efficient to implement: given a pre-trained class probability model (e.g. logistic regression), the algorithm performs a sequence of efficient threshold optimizations on the predicted class probability outputs. Secondly, it can be applied readily to multi-class problems and fairness problems with multiple groups. Thirdly, the number of optimization parameters needed by our algorithm scales linearly with the number of classes and groups, and does not directly depend of the number of constraints. This is in contrast to the method of Narasimhan (2018), which maintains an explicit parameter for each constraint. Our approach instead solves a linear minimization problem over the feasible matrices, which has the advantage of leveraging specialized convex solvers that exploit redundancies in the constraints.

Contributions. The following are the main contributions in this paper. (i) We provide an algorithm for complex constrained classification problems , which solves a sequence of plug-in learning tasks (see Section 3). (ii) We show that our algorithm is statistically consistent and enjoys improved convergence guarantees (see Section 4). (iii) We present experiments on benchmark fairness datasets and show that the proposed algorithm performs at least as well as existing methods, while being more robust to choices of hyper-parameters (see Section 5).

Related Work. Prior methods for optimizing complex evaluation metrics fall mainly under two broad categories: plug-in style methods that enjoy consistency guarantees [35, 25, 34, 33, 44, 3, 29], and approaches that optimize convex relaxations to the metrics and are not necessarily consistent [20, 22, 32, 21, 16, 37, 30, 19]. There has also been much work on training classifiers with objectives and constraints that are *linear* constraints on the confusion matrix, with the main focus being on fairness constraints [16, 46, 2, 23, 11, 9, 10, 31]. There's however been relatively lesser work on handling objectives and constraints that are non-linear in the confusion matrix [29, 30, 5]. The more recent of these approaches by Narasimhan et al. (2019) [30] formulates the constrained learning problem as a Lagrangian game played by three players, and seeks to find an equilibrium of the game. However, their main proposal makes use of "surrogate relaxations" for the entries of the confusion matrix and does not come with consistency guarantees. We compare against this algorithm in our experiments. Narasimhan et al. (2019) do however also provide a more idealized algorithm that enjoys the same convergence rate as our approach to the optimal feasible solution, but do not provide a consistency analysis for this method. In Section 4 and Appendix B, we discuss in detail about the technical differences between this idealized algorithm of theirs and our approach.

2 Preliminaries and Background

We are interested in general multiclass learning problems with an instance space $\mathcal{X}$ and label space $\mathcal{Y} = [n] = \{1, 2, \ldots, n\}$. For binary classification problems, we will denote the label space using $\mathcal{Y} = \{0, 1\}$. We use Δ_n to denote the probability simplex in $\mathbb{R}^n_+$. We assume examples are drawn i.i.d. from some distribution D on $\mathcal{X} \times [n]$, with marginal μ on $\mathcal{X}$, conditional class probabilities $\eta_i(x) = \mathbf{P}(Y = i|X = x)$, and class priors $\pi_i = \mathbf{P}(Y = i)$. Given a finite training sample $S = ((x_1, y_1), ..., (x_N, y_N)) \in (\mathcal{X} \times [n])^N$ drawn i.i.d. from D, the task 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 $h(x)$. Let $\mathcal{H}$ denote the the space of all randomized classifiers.

We will also be interested in fair classification problems where each instance belongs to one of m protected groups, and will denote the protected group associated with instance X by $A(X) \in [m]$. We denote $\nu_a = \mathbf{P}(A(X) = a)$ and $\pi_{a,i} = \mathbf{P}(A(X) = a, Y = i)$.

Learning problem. We measure the performance of a classifier w.r.t. distribution D using a performance measure $\bar{\psi} : \mathcal{H} \rightarrow R_+$ that associates a non-negative value $\bar{\psi}(h; D) \in \mathbb{R}_+$ to each classifier $h \in \mathcal{H}$, with *lower* values indicating better performance. We also require the classifier to satisfy K constraints, given by $\bar{\phi}_k(h; D) \leq 0, k \in [K]$, where $\bar{\phi}_k : \mathcal{H} \rightarrow \mathbb{R}$ associates a real value to a classifier. Our goal is to then solve the following optimization problem over classifiers:

$$\min_{h \in \mathcal{H}} \bar{\psi}(h) \;\text{ s.t. }\; \bar{\phi}_k(h) \leq 0,\, \forall k \in [K]. \tag{OP1}$$

Confusion matrices. We define the confusion matrix of a classifier h as a $n \times n$ matrix $C[h] \in [0, 1]^{n \times n}$ where the ij-the entry is the probability that the true class for an instance is i and the predicted class is j:

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

where $\widehat{Y} \sim h(X)$ denotes a random draw of label from $h(X)$. For fairness settings, we will also be interested in the group-specific confusion matrices:

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

Complex objectives and constraints. We will consider performance metrics $\bar{\psi}$ and constraint functions $\bar{\phi}_k$'s that are general functions of the confusion matrix of classifier h. This includes several common examples, including those that are *non-decomposable* and cannot be expressed as a simple expectation of errors on individual examples.

- *Class-imbalanced metrics* such as the G-mean, H-mean and Q-mean that emphasize equal performance acrossa all classes [27, 24, 39, 42, 26, 28] and metrics used in signal detection [41]:

$$\text{G-mean} \;=\; 1 - \left(\textstyle\prod_{i=1}^n \frac{C_{ii}}{\pi_i}\right)^{1/n}; \qquad \text{H-mean} \;=\; 1 - n\left(\textstyle\sum_{i=1}^n \frac{\pi_i}{C_{ii}}\right)^{-1}$$

$$\text{Q-mean} \;=\; \sqrt{\tfrac{1}{n}\textstyle\sum_{i=1}^n \left(1 - \frac{C_{ii}}{\pi_i}\right)^2}; \qquad \text{Min-max} \;=\; \max_{i \in [n]} \left(1 - \tfrac{C_{ii}}{\pi_i}\right)$$

- *Fairness constraints* used to control the discrepancy in the performance of a classifier across different protected groups [17]:

$$\text{Demographic Parity:}\;\; \max_{a \in [m]} \left|\tfrac{1}{\nu_a}(C^a_{01} + C^a_{11}) - \tfrac{1}{m}\textstyle\sum_{b=1}^m \frac{1}{\nu_b}\left(C^b_{01} + C^b_{11}\right)\right| \leq \epsilon$$

$$\text{Equal Opportunity:}\;\; \max_{a \in [m]} \left|\tfrac{1}{\pi_{a,1}} C^a_{11} - \tfrac{1}{m}\textstyle\sum_{b=1}^m \frac{1}{\pi_{b,1}} C^b_{11}\right| \leq \epsilon,$$

  where ϵ is an acceptable slack.

- *Coverage constraints* that require the proportion of predictions in a particular class to match a target value [16, 10, 8], and the related KL-divergence metric used in the quantification literature [12, 14, 21]:

$$\text{Binary Coverage:}\;\; C_{01} + C_{11} \leq \epsilon$$

$$\text{KL-divergence:}\;\; \textstyle\sum_{i=1}^n \pi_i \log\left(\frac{\pi_i}{\sum_{j=1}^n C_{ji}}\right) \leq \epsilon.$$

Confusion vectors. For ease of presentation, we will work with a generalized version of a confusion matrix, which we refer to as a confusion vector. For a classifier h, we overload notation and define a confusion vector $C[h] \in \mathbb{R}^d$ as:

$$C_i[h] = \mathbf{E}_{X,Y}[\mathbf{E}_{\widehat{Y} \sim h(X)}[\sigma_i(X, Y, \widehat{Y})]],$$

for some *sufficient statistics* $\sigma_i : \mathcal{X} \times [n] \times [n] \rightarrow [0, 1]$ computed on the instance X, true labels Y and predicted labels $\widehat{Y}$. For example, when $\sigma_i(X, Y, \widehat{Y}) = \mathbf{1}(Y = i, \widehat{Y} = i)$, we get the diagonal elements of the standard confusion matrix with $d = n$. When we set $\sigma_i(X, Y, \widehat{Y}) = \mathbf{1}(Y = j, \widehat{Y} = k)$, we get back the jk-th entry of the standard confusion matrix, with the entire matrix can be represented by a n^2-dimensional confusion vector. When we set $\sigma_i(X, Y, \widehat{Y}) = \mathbf{1}(A(X) = a, Y = j, \widehat{Y} = k)$, we get back the jk-th entry of the group-specific confusion matrix for group a. The set of m group-specific matrices can then be represented by a mn^2-dimensional confusion vector.

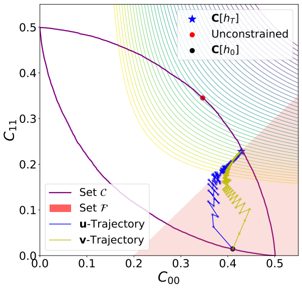


Figure 1: An illustration of Algorithm 1 for a toy 2-class problem, with equal prior probabilities and with class conditionals $X|Y = 0$ and $X|Y = 1$ distributed as a standard normal with means $+1$ and -1 respectively. The goal is to minimize H-mean subject to a coverage constraint that forces the fraction of class 1 predictions to be not more than 0.3. The objective and constraint functions are given by: $\psi(C) = 1 - 2\big(\frac{0.5}{C_{00}} + \frac{0.5}{C_{11}}\big)^{-1}$ and $\phi(C) = C_{11} + C_{01} - 0.3$.

3 Reduction-based Algorithm

We now describe our approach for solving the learning problem in (OP1) by reducing the problem into a sequence of plug-in classifier learning tasks. We will work with objectives and constraints defined in terms of a confusion vector $C[h]$ of dimension d, for some suitable choice of sufficient statistics σ_i's. Specifically, we consider an objective $\bar{\psi}(h) = \psi(C[h])$ defined by a *convex* function $\psi : [0,1]^d \to \mathbb{R}$ of the confusion vector for h, and constraint functions $\bar{\phi}_k(h) = \phi_k(C[h])$ defined by convex functions $\phi_k : [0,1]^d \to \mathbb{R}$ of the confusion vector for h.

3.1 Optimization Over Intersection of Convex Sets

Our key idea is to reformulate (OP1) as an optimization problem over the intersection of two convex sets. To this end, we define the set of all confusion vectors that can be achieved by some classifier h:

$$\textbf{Achievable Confusion Vectors: } \mathcal{C} = \{\mathbf{u} \in \mathbb{R}^d : \mathbf{u} = C[h], h : \mathcal{X} \to \Delta_n\},$$

and the set of confusion vectors that satisfy the K constraints:

$$\textbf{Feasible Confusion Vectors: } \mathcal{F} = \{\mathbf{u} \in \mathbb{R}^d : \phi_k(\mathbf{u}) \leq 0, \, \forall k \in [K]\}.$$

Proposition 1. *$\mathcal{C}$ and $\mathcal{F}$ are convex sets.*

The convexity of $\mathcal{C}$ follows from the use of randomised classifiers and the fact that $C[h]$ is defined as an expectation over random draw from h. The convexity of $\mathcal{F}$ follows from the convexity of the constraint functions ϕ_k. Also notice that while the set of achievable confusion vectors $\mathcal{C}$ depends on the data distribution D, the set of feasible confusion vectors does not. This means that optimizing over $\mathcal{F}$ does not require access to D or a sample drawn form D.

Equipped with these two sets, we can reformulate the learning problem in (OP1) over the space of classifiers, as an equivalent d-dimensional optimization problem over the intersection of $\mathcal{C}$ and $\mathcal{F}$:

$$\min_{\mathbf{u} \in \mathcal{C} \cap \mathcal{F}} \psi(\mathbf{u}). \tag{OP2}$$

We will denote the solutions to the problems (OP1) and (OP2) by h^* and $\mathbf{u}^*$ respectively. Note that $C[h^*] = \mathbf{u}^*$. In Figure 1, we provide a simple illustration of an objective function and constraints on a toy problem, and show the corresponding sets $\mathcal{C}$ and $\mathcal{F}$.

3.2 Linear Minimization Oracles

The formulation (OP2) converts a classifier learning problem into a finite dimensional optimization problem, but it still has one major issue: *we do not have direct access to the set $\mathcal{C}$*. However, as we shall see shortly, performing a linear minimization over this set amounts to a cost-sensitive learning problem, which can be solved using a plug-in method. Similarly, performing a linear minimization over $\mathcal{F}$ amounts to solving a convex program.

So, we assume access to the following linear minimization oracles (LMOs):

$$\text{LMO}_{\mathcal{C}}\text{: Given } \mathbf{a} \in \mathbb{R}^d \text{, returns } \operatorname*{argmin}_{\mathbf{u} \in \mathcal{C}} \langle \mathbf{a}, \mathbf{u} \rangle,$$
$$\text{LMO}_{\mathcal{F}}\text{: Given } \mathbf{b} \in \mathbb{R}^d \text{, returns } \operatorname*{argmin}_{\mathbf{v} \in \mathcal{F}} \langle \mathbf{b}, \mathbf{v} \rangle.$$

Algorithm 1 The Split Bayes-Frank-Wolfe (SBFW) Algorithm

```
Input: $\psi : [0,1]^d \to \mathbb{R}_+$, Linear minimization oracle over $\mathcal{F}$
        Training sample $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$
Parameters: $\lambda > 0$, Step sizes $\eta_t = C/t$, and $\gamma_t = \frac{4\eta_t}{\lambda}$ for $t \in [T]$, where $C$ is some constant.
Initialize: Initialize classifier $h_0 : \mathcal{X} \to \Delta_n$ and vectors $\mathbf{u}_0 = \mathbf{v}_0 = C[h_0]$, $\mathbf{w}_0 = 0$.
For $t = 1$ to $T$ do:
    $\widehat{g}_t, \widetilde{\mathbf{u}}_t = \text{plug-in}(\mathbf{a}_{t-1}; S)$, where $\mathbf{a}_{t-1} = \nabla_{\mathbf{u}} \mathcal{L}(\mathbf{u}_{t-1}, \mathbf{v}_{t-1}, \mathbf{w}_{t-1})$    (LMO over $\mathcal{C}$)
    $\widetilde{\mathbf{v}}_t = \text{argmin}_{\mathbf{v} \in \mathcal{F}} \langle \mathbf{b}_{t-1}, \mathbf{v} \rangle$, where $\mathbf{b}_{t-1} = \nabla_{\mathbf{v}} \mathcal{L}(\mathbf{u}_{t-1}, \mathbf{v}_{t-1}, \mathbf{w}_{t-1})$    (LMO over $\mathcal{F}$)
    $(\mathbf{u}_t, \mathbf{v}_t, h_t) = (1 - \gamma_t)(\mathbf{u}_{t-1}, \mathbf{v}_{t-1}, h_{t-1}) + \gamma_t(\widetilde{\mathbf{u}}_t, \widetilde{\mathbf{v}}_t, \widehat{g}_t)$
    $\mathbf{w}_t = \mathbf{w}_{t-1} + \eta_{t-1}(\mathbf{u}_t - \mathbf{v}_t)$
end For
Return: Return $\widehat{h} = h_{\text{Best}}$, where Best = $\text{argmin}_{t > T/2} ||\mathbf{u}_t - \mathbf{v}_t||^2$
```

Of the two oracles, LMO$_{\mathcal{F}}$ does not need access to the data and can performed with standard convex solvers. So, we will be primarily interested in the number of calls needed to be made to LMO$_{\mathcal{C}}$. Also note that in practice, one may not be able to solve the minimization over $\mathcal{C}$ exactly. In our theoretical analysis in Section 4, we take this into account and show that our approach is robust to approximation errors in the linear minimization.

3.3 Frank-Wolfe Based Algorithm

The challenge now is to optimize over the intersection of the two sets $\mathcal{C} \cap \mathcal{F}$. For this, we adopt the Frank-Wolfe based approach of Gidel et al. (2018) [15] that enables optimization of a convex objective over the intersection of two convex sets with access to only linear minimization oracles for the individual sets. To this end, we introduce auxiliary variables $\mathbf{v}$ in (OP2) and decouple the two constraint sets. This gives us the following equivalent optimization problem:

$$\min_{\mathbf{u} \in \mathcal{C}, \mathbf{v} \in \mathcal{F}} \psi(\mathbf{u}) + \psi(\mathbf{v}) \;\; \text{s.t.} \;\; \mathbf{u} - \mathbf{v} = 0. \tag{OP3}$$

We then define the augmented Lagrangian $\mathcal{L} : [0,1]^d \times [0,1]^d \times \mathbb{R}^d \to \mathbb{R}$ of the above problem as:

$$\mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \psi(\mathbf{u}) + \psi(\mathbf{v}) + \mathbf{w}^\top(\mathbf{u} - \mathbf{v}) + \frac{\lambda}{2}||\mathbf{u} - \mathbf{v}||^2, \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^d$ denotes the Lagrange multipliers for the equality constraints and $\lambda > 0$ is a constant.

Gidel at al. (2018) [15] propose a simple gradient ascent step for $\mathbf{w}$, a linear minimization step for $\mathbf{u}$ over $\mathcal{C}$ and a linear minimization step for $\mathbf{v}$ over $\mathcal{F}$. Specifically, at each iteration, we perform a Frank-Wolfe style update for $\mathbf{u}$ and $\mathbf{v}$ [18]. We linearize the Lagrangian with respect to $\mathbf{u}$ and minimize the linearized objective over $\mathcal{C}$ using LMO$_{\mathcal{C}}$:

$$\mathbf{a}_{t-1} = \nabla_{\mathbf{u}} \mathcal{L}(\mathbf{u}_{t-1}, \mathbf{v}_{t-1}, \mathbf{w}_{t-1}); \;\; \widetilde{\mathbf{u}}_t \in \text{argmin}_{\mathbf{u} \in \mathcal{C}} \langle \mathbf{a}_{t-1}, \mathbf{u} \rangle. \tag{2}$$

We also linearize $\mathcal{L}$ with respect to $\mathbf{v}$ and minimize the linearized objective over $\mathcal{F}$ using LMO$_{\mathcal{F}}$:

$$\mathbf{b}_{t-1} = \nabla_{\mathbf{v}} \mathcal{L}(\mathbf{u}_{t-1}, \mathbf{v}_{t-1}, \mathbf{w}_{t-1}); \;\; \widetilde{\mathbf{v}}_t \in \text{argmin}_{\mathbf{v} \in \mathcal{F}} \langle \mathbf{b}_{t-1}, \mathbf{v} \rangle. \tag{3}$$

This is followed by a set of simple updates on the optimization variables:

$$\mathbf{u}_t \; = (1 - \gamma_t)\mathbf{u}_{t-1} + \gamma_t \widetilde{\mathbf{u}}_t; \qquad \mathbf{v}_t \; = (1 - \gamma_t)\mathbf{v}_{t-1} + \gamma_t \widetilde{\mathbf{v}}_t; \tag{4}$$

$$\mathbf{w}_t \; = \mathbf{w}_{t-1} + \eta_{t-1}(\mathbf{u}_t - \mathbf{v}_t), \tag{5}$$

where the coefficients γ_t and η_t are step-size parameters. The procedure outlined in Algorithm 1 maintains both a confusion vector and the corresponding classifier at each iteration, and returns a classifier $\widehat{h}$ that combines multiple classifiers via randomization.

3.4 Plug-in Classifier for LMO over $\mathcal{C}$

All that remains is to perform the linear minimization over $\mathcal{C}$ in Equation 2. We show below that this can be solved using a plug-in method.

Algorithm 2 Plug-in Method for LMO$_\mathcal{C}$

```
Input: Weight vector $\mathbf{a} \in \mathbb{R}^d$, Training sample $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$
Given: A conditional probability model $\widehat{\eta} : \mathcal{X} \to \Delta_n$ pre-trained with samples $\{(x_i, y_i)\}_{i=1}^{N/2}$,
    Sufficient statistic functions $\sigma_i : \mathcal{X} \times [n] \times [n] \to [0, 1]$
Define $\mathbf{L} : \mathcal{X} \to \mathbb{R}^{n \times n}$ by $L_{j,k}(x) = \sum_{i=1}^{d} a_i \sigma_i(x, j, k)$
Construct $\widehat{g} : \mathcal{X} \to [n]$ as $\widehat{g}(x) = \operatorname{argmin}_{\widehat{y} \in [n]} \sum_{j=1}^{n} \widehat{\eta}_j(x)\, L_{j,\widehat{y}}(x)$,
Estimate confusion vector $\widetilde{u}_i = \frac{2}{N} \sum_{j=N/2}^{N} \sigma_i(x_j, y_j, \widehat{g}(x_j))$ from samples $\{(x_i, y_i)\}_{i=N/2}^{N}$
Return: Confusion vector $\widetilde{\mathbf{u}}$ and corresponding classifier $\widehat{g}$
```

Proposition 2 (LMO$_\mathcal{C}$ through Bayes-optimal Classifier). *Suppose we wish to minimize $\langle \mathbf{a}, \mathbf{u} \rangle$ over $\mathbf{u} \in \mathcal{C}$. Define the example-dependent loss matrix $\mathbf{L} : \mathcal{X} \to \mathbb{R}^{n \times n}$ as $L_{j,k}(x) = \sum_{i=1}^{d} a_i \sigma_i(x, j, k)$. Then the solution to the linear minimization problem is directly given by the Bayes-optimal classifier for this loss matrix. Specifically, construct a classifier $g^* : \mathcal{X} \to [n]$ with*

$$g^*(x) = \operatorname*{argmin}_{\widehat{y} \in [n]} \sum_{j=1}^{n} \eta_j(x)\, L_{j,\widehat{y}}(x),$$

where $\eta_j(x) = \mathbf{P}(Y = 1|x)$ is the class-conditional probability. Then $C[g^*] \in \operatorname{argmin}_{\mathbf{u} \in \mathcal{C}} \langle \mathbf{a}, \mathbf{u} \rangle$.

The classifier g^* defined above is a deterministic classifier that thresholds the conditional probability η based on the example-dependent loss matrix $\mathbf{L}(x)$. For the special case where the confusion vectors represent the set of confusion matrices for the m groups, the weight vector $\mathbf{a} \in \mathbb{R}^{mn^2}$ effectively describes m loss matrices $\mathbf{L}^1, \ldots, \mathbf{L}^m \in \mathbb{R}^{n \times n}$, one for each group. For a given instance x, the classifier g^* picks the loss matrix $\mathbf{L}^{A(x)}$ associated with the protected group attribute $A(x)$, and then uses the conditional probability vector $\eta(x)$ to make the optimal prediction for that loss matrix: $g^*(x) = \operatorname{argmin}_{\widehat{y} \in [n]} \sum_{j=1}^{n} \eta_j(x)\, L_{j,\widehat{y}}^{A(x)}(x)$.

The above characterization directly motivates the use of a plug-in method to solve the LMO over $\mathcal{C}$. Specifically, we can use an estimator $\widehat{\eta} : \mathcal{X} \to \Delta_n$ of the conditional probabilities η to construct an approximate version of g^*. The confusion vector $C[g^*]$ can then be estimated from samples. This procedure is outlined in Algorithm 2 and returns both a confusion vector that approximately solves the linear minimization over $\mathcal{C}$ and the corresponding classifier $\widehat{g}$. Notice that the conditional probability estimator $\widehat{\eta}$ (e.g. logistic regression) needs to be trained only once, and can be re-used for every call to the plug-in routine.

Figure 1 shows the iterates of the proposed algorithm over a simple toy dataset. The trajectory of $\mathbf{u}_t$ is given in blue and the trajectory of $\mathbf{v}_t$ is given in yellow. It can be seen that both these trajectories approach the optimal solution $C[h^*]$.

4 Consistency Results

In this section we give the main theoretical result of the paper. We show that with $O(1/\epsilon^2)$ calls to the plug-in LMO routine, Algorithm 1 outputs a classifier $\widehat{h}$ that is $O(\epsilon + \sqrt{\rho})$-close to the optimal-$\psi$ value and satisfies the constraint ϕ_k's with a slack of $O(\epsilon + \sqrt{\rho})$, where ρ is a term which depends on the approximation level of the plug-in LMO. This result then directly implies that Algorithm 1 is statistically consistent, i.e. converges to the optimal-feasible classifier in the limit of infinite samples.

We will make a few regularity assumptions. We assume that the objective function ψ and constraint functions ϕ_k are L-Lipschitz and objective function ψ is β-smooth. We will also assume that (OP2) is strictly feasible.

Assumption 1. $\exists \mathbf{u} \in \mathcal{C} \cap \mathcal{F}, r > 0$ *such that* $B(\mathbf{u}, r) \cap$ *affine-space*$(\mathcal{C}) \subseteq \mathcal{C} \cap \mathcal{F}$.

We stress that these assumptions are not very restrictive and can be verified to be satisfied by all of the objectives and constraints described in Section 2, as long as the prior probabilities $\pi_{a,i}$ are non-zero for all classes $i \in [n]$ and protected groups $a \in [m]$.

Theorem 3. *Let h^* denote the optimal feasible solution for* (OP1)*, i.e.* $\phi_k(C[h^*]) \leq 0, \forall k$ *and* $\psi(C[h^*]) \leq \psi(C[h])$ *for all h that is feasible. Under the regularity assumptions, for large enough λ and an appropriate step-size parameter C, there exists an $\bar{\epsilon} > 0$ such that, for all $\epsilon \leq \bar{\epsilon}$, and $T \geq \frac{c}{\epsilon^2}$, with probability $1 - \delta$ over draw of the training samples S i.i.d. from D, the classifier $\widehat{h}$ returned by Algorithm 1 is near-optimal and near-feasible:*

$$\textbf{\textit{Optimality}}: \;\; \psi(C[\widehat{h}]) \; \leq \psi(C[h^*]) \; + \; c\sqrt{\rho} + \epsilon,$$
$$\textbf{\textit{Feasibility}}: \;\; \phi_k(C[\widehat{h}]) \; \leq c\sqrt{\rho} + L\epsilon, \; \forall k \in [K],$$

where $\rho = \sqrt{d}\mathbf{E}||\eta(X) - \widehat{\eta}(X)||_1 + d\sqrt{\frac{d\log(d)+\log(Nn^2)+\log(1/\delta)}{N}}$ captures the approximation level of the LMO given by Algorithm 2, and $c > 0$ is a constant not dependent on the number of iterations T and the training samples.

The key to proving this convergence result is (i) establishing that the plug-in classifier solves the linear minimization problem over $\mathcal{C}$ approximately, (ii) applying the convergence results of Gidel et al. (2018) [15] (extended to handle an approximate LMO) to get a bound on the duality gap for (OP2), and (iii) translating this to a bound on the optimality and feasibility for (OP2).

Remark (Consistency). The term ρ in Theorem 3 has two sources of error: the error $\mathbf{E}||\eta(X) - \widehat{\eta}(X)||_1$ in the class probability model $\widehat{\eta}$ used to construct the plug-in classifier and the sample error $\widetilde{\mathcal{O}}\Big(d\sqrt{\frac{d}{N}}\Big)$. If the conditional-class estimator is such that $\mathbf{E}||\eta(X) - \widehat{\eta}(X)||_1 \to 0$ as the sample size $N \to \infty$, which is the case when e.g. $\widehat{\eta}$ is learned by minimizing a strictly proper composite loss over a suitably flexible function class [40], then Algorithm 1 is guaranteed to be statistically consistent. Specifically, setting $\epsilon = \sqrt{1/N}$ and running Algorithm 1 for the prescribed $O(1/\epsilon^2)$ iterations, we have that as $N \to \infty$, $\psi(C[\widehat{h}]) \xrightarrow{P} \psi(C[h^*])$ and $\phi_k(C[\widehat{h}]) \xrightarrow{P} 0, \forall k$.

Remark (Improvements over COCO [29]). The previous reduction-based algorithm of Narasimhan (2018) [29] for (OP1), referred to as COCO by the author, similarly poses the problem as an optimization over $\mathcal{C}$ but retains explicit constraints $\phi_k(C) \leq 0, \forall k$. The idea is to then formulate the Lagrangian for the constrained problem with one Lagrange multiplier for each constraint, and maximize the Lagrangian over the multipliers using gradient ascent. Each gradient step, however, involves a full run of the classical Frank-Wolfe method [18] over $\mathcal{C}$ using an LMO, resulting in an algorithm with multiple levels of nesting. Our approach is better than COCO in the following aspects:

- *Better convergence rate*. In the large N setting, COCO requires $O(1/\epsilon^3)$ calls to the plug-in routine to reach a solution that is $O(\epsilon)$-optimal and $O(\epsilon)$-feasible. In contrast, by posing (OP1) as an optimization over two convex sets, we avoid the nested structure, and need only $O(1/\epsilon^2)$ calls to the plug-in routine to reach a solution of the same quality.
- *Weaker dependence on the number of constraints*. While COCO maintains one optimization parameter per constraint, the number of parameters in our algorithm (i.e. $\mathbf{u}$, $\mathbf{v}$) is only twice the dimension d of the confusion vector, and depends on the number of constraints K only through the LMO over $\mathcal{F}$. This has the added advantage of being able to use specialized solvers for this step that better exploit the redundancies in the constraint set.

Remark (Prior 3-player approach [30]). As noted in the introduction, another closely related method for solving complex constrained classification problems is the 3-player approach of Narasimhan et al. (2019) [30]. Their idea is to introduce additional slack variables, formulate the Lagrangian for the problem with one parameter per constraint, and find an equilibrium of the resulting min-max game between the primal and dual variables. They first provide an idealized version of their algorithm which makes use of an oracle (similar to LMO$_{\mathcal{C}}$) to optimize a linear metric over the space of classifiers, and requires a similar number of calls to the oracle as our approach to reach a near-optimal near-feasible solution. However, they do not provide a full-fledged consistency analysis for this idealized algorithm. Instead they prescribe a "practical" alternative which replaces the oracle with stochastic gradient updates on a relaxed Lagrangian, where the entries of the confusion matrix are replaced with surrogate relaxations, and this variant does not come with consistency guarantees. We compare with this surrogate-based algorithm in our experiments. Again, an important difference between our approach and Narasimhan et al. (2019) is that we do not maintain an explicit parameter for each constraint and access the constraint set only through an LMO.

Table 1: Minimizing Q-mean s.t. Demographic Parity ≤ 0.05. We report test Q-mean and constraint violations (in parentheses) measured as the positive part of Demographic Parity $- 0.05$. *Lower* values are better. **Bold** indicates that the method has the least objective and the least violation, among the last three columns.

Dataset	Unconstrained	Error-Con	COCO	3-Player	Proposed
Adult	0.18 (0.05)	0.30 (0.00)	0.31 (0.00)	**0.18 (0.00)**	**0.18 (0.00)**
COMPAS	0.32 (0.10)	0.36 (0.00)	0.35 (0.03)	0.33 (0.00)	**0.32 (0.00)**
Crimes	0.16 (0.22)	0.30 (0.01)	0.30 (0.01)	0.24 (0.05)	**0.22 (0.03)**
Default	0.33 (0.01)	0.54 (0.00)	0.35 (0.00)	0.36 (0.00)	**0.33 (0.00)**
Lawschool	0.21 (0.25)	0.47 (0.00)	0.35 (0.16)	0.24 (0.03)	0.25 (0.02)

Table 2: Minimizing G-mean s.t. Equal Opportunity ≤ 0.05. We report G-mean and constraint violations measured as the positive part of Equal Opportunity $- 0.05$. *Lower* values are better.

Dataset	Unconstrained	Error-Con	COCO	3-Player	Proposed
Adult	0.18 (0.00)	0.24 (0.01)	0.17 (0.03)	0.18 (0.01)	0.18 (0.00)
COMPAS	0.32 (0.09)	0.35 (0.00)	**0.32 (0.00)**	0.33 (0.00)	**0.32 (0.00)**
Crimes	0.15 (0.17)	0.19 (0.09)	0.16 (0.06)	0.16 (0.08)	**0.16 (0.03)**
Default	0.33 (0.00)	0.51 (0.00)	0.39 (0.00)	0.36 (0.00)	**0.34 (0.00)**
Lawschool	0.21 (0.23)	0.47 (0.00)	0.23 (0.00)	0.22 (0.04)	0.26 (0.03)

We provide more details about the prior COCO and 3-player methods in Appendix B.

5 Experiments

We show that the proposed algorithm performs comparable to or better than than prior methods for constrained classification on a number of benchmark datasets for fair classification.

Datasets. We ran experiments on five datasets: (1) *COMPAS*, where the goal is to predict recidivism with *gender* as the protected attribute [4]; (2) *Communities & Crime*, where the goal is to predict if a community in the US has a crime rate above the 70th percentile [13], and we consider communities having a black population above the 50th percentile as protected [23]; (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 [43]; (4) *Adult*, where the task is to predict if a person's income exceeds 50K/year, with *gender* as the protected attribute [13]; (5) *Default*, where the task is to predict if a credit card user defaulted on a payment, with gender as the protected attribute [13]. The details are summarized in Table 4 in Appendix C. We used 2/3-rd of the data for training and 1/3-rd for testing. All experiments use a linear model.[1]

Comparisons. We compare our method against (i) the approach of optimizing the given objective without constraints [33] (Unconstrained), (ii) the approach of optimizing classification error subject to the given constraints, e.g. [1] (Error-Con), (iii) the prior COCO method [29] for solving the constrained learning problem at hand, and (iv) the 3-player approach [30] which solves the constrained learning problem with surrogates. We describe how we choose hyper-parameters in Appendix C

Objectives and Constraints. We consider the following constrained learning tasks:

1. Minimizing Q-mean s.t. Demographic Parity Violation ≤ 0.05
2. Minimizing G-mean s.t. Equal Opportunity Violation ≤ 0.05
3. Minimizing H-mean s.t. Coverage for Class $1 \leq 0.25$

We report the objectives and constraint violations (the positive part of $\phi(h) - \epsilon$) for the different methods in Tables 1–3. On a majority of the datasets, the proposed method is able to closely satisfy the constraints while achieving comparable or better objectives. As expected, unconstrained optimization of the objective performs poorly on the constraints. Similar, optimizing for plain error rate subject to the specified constraints fares poorly on the desired objective, demonstrating the need to directly optimize for the metric one cares about. Among the SBFW (proposed), COCO and 3-Player methods, our approach is able to more often achieve the least objective and the least violation.

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

Table 3: Minimizing H-mean s.t. Class-1 Coverage ≤ 0.25. We report etst H-mean and constraint violations measured as the positive part of Coverage $- 0.25$. *Lower* values are better. **Bold** indicates that the method has the least objective and the least violation, among last three columns.

Dataset	Unconstrained	Error-Con	COCO	3-Player	Proposed
Adult	0.18 (0.09)	0.26 (0.00)	**0.21 (0.00)**	**0.21 (0.00)**	**0.21 (0.00)**
COMPAS	0.32 (0.23)	0.44 (0.00)	0.45 (0.00)	0.45 (0.00)	**0.44 (0.00)**
Crimes	0.16 (0.11)	0.21 (0.01)	0.21 (0.01)	0.23 (0.00)	0.21 (0.01)
Default	0.33 (0.16)	0.62 (0.00)	**0.34 (0.00)**	0.42 (0.00)	**0.34 (0.00)**
Lawschool	0.21 (0.47)	0.56 (0.01)	0.58 (0.00)	0.56 (0.00)	0.55 (0.01)

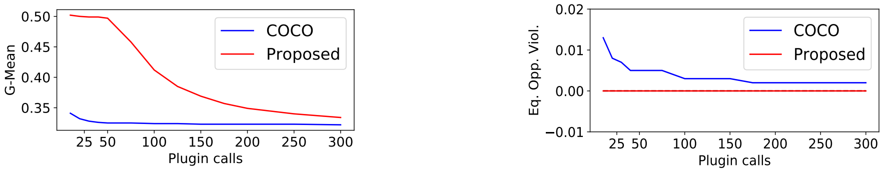


Figure 2: Training G-mean (left) and equal opportunity violation (right) on COMPAS for varying number of calls to the plug-in routine. The hyper-parameters were tuned separately for each method using the heuristic of Cotter et al. (2019) [10] to trade-off between the objective and the violations.

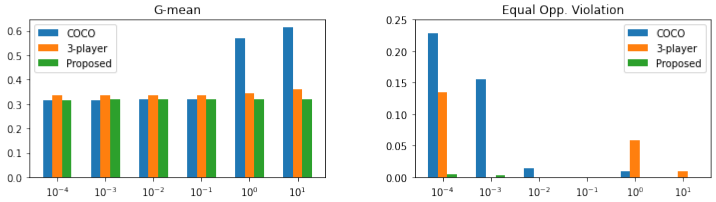


Figure 3: Robustness to hyper-parameters: Train G-mean and equal opportunity violation for six step sizes (*lower* is better) on the COMPAS dataset. For the proposed algorithm, all choices achieved similar objectives and near-zero violations.

Convergence Analysis. We next compare the number of plug-in calls needed by the proposed algorithm and the previous COCO method for the task of minimizing G-mean with an equal opportunity constraint. The 3-player method does not use a plug-in subroutine. Figure 2 shows the train G-mean and the train equal opportunity violation (the positive part of $\phi(h) - \epsilon$) for varying numbers of plug-in calls for the COMPAS dataset. In this case, our algorithm converges to a classifier with zero violation on the training set, with an objective similar to COCO, but with fewer calls (≤ 100). We also provide similar plots for other datasets in Figure 4 in Appendix C. On Crimes and Law School, COCO fails to converge to zero training violation even after 2000 calls. In contrast, on all five datasets, when provided the same number of plug-in calls, the proposed algorithm is able to achieve zero training violations (often within the first 100 calls). On Adult alone, COCO exhibits faster convergence.

Robustness to Hyper-parameter Choices. In our final experiment, we demonstrate the robustness of our approach to the choice of step-size η_t. We ran COCO, 3-player and the proposed SBFW methods for minimizing G-mean objective with an equal opportunity constraint on the COMPAS dataset, with 6 different choices of step-sizes $(10^{-4}, 10^{-3}, \ldots, 10)$, and report the G-mean and equal opportunity violation in Figure 3 (and also as a scatter plot in Figure 5 in the Appendix). While all 6 choices achieved close-to-best objectives and near-zero violations for the proposed SBFW algorithm, only 2 (3 resp.) choices led to similar metrics for COCO (3-player resp.).

6 Conclusion

In numerous real-word prediction tasks, one is required to learn a classifier that optimizes a complex evaluation metric subject to a set of constraints. In this paper, we developed a consistent learning algorithm for handling objectives and constraints that are convex functions of the confusion matrix and provided improved convergence guarantees. In our experiments, we demonstrated the effectiveness of our approach, and also showed its robustness to hyper-parameter choices. In the future, it would be interesting to explore lower bounds on the number of calls to the LMO, replace the plug-in

LMO routine with more direct cost-sensitive learning methods (e.g. [38, 45]), and explore other optimization methods in place of the augmented Lagrangian Frank-Wolfe algorithm.

Broader Impact

There's an increasing impetus in the machine learning community to design algorithms that are fair and free from bias and inequity. Most existing approaches for enforcing group-based fairness goals have been limited to simple objectives and constraints. In this paper, we allow a user to specify for more nuanced definitions of utilities and fairness goals than allowed by standard methods in the literature, and provide an algorithm to directly and efficiently optimize for these goals. We show theoretically that our algorithm is able to achieve a desired trade-off between overall utility and the specified fairness criteria.

As with prior work on group-based fairness (and more generally with constrained supervised learning), a drawback of our approach is that while we guarantee that the fairness criterion is likely to be satisfied on new examples, there is a small probability that it isn't, and these rare failures can have an adverse impact in practice. Moreover, our algorithm requires the use of stochastic classifiers, which may bring in additional ethical considerations. See Cotter et al. [8] for a discussion on the practical ramifications of deploying a stochastic classifier, and for ways to convert a stochastic classifier into a similar performing deterministic classifier.

All experiments in this paper were carried out with publicly available datasets.

Acknowledgments and Disclosure of Funding

SKT and HGR thank the Robert Bosch Center for Data Science and Artificial Intelligence at IIT Madras for their support. The authors thank Shivani Agarwal for helpful discussions.

References

[1] A. Agarwal, A. Beygelzimer, M. Dudik, and J. Langford. A reductions approach to fair classification. In *FAT/ML*, 2017.

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

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

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

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

[6] N. Cesa-Bianchi and D. Haussler. A graph-theoretic generalization of the sauer-shelah lemma. *Discrete Applied Mathematics*, 1998.

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

[8] A. Cotter, M. Gupta, and H. Narasimhan. On making stochastic classifiers deterministic. In *Advances in Neural Information Processing Systems*, pages 10912–10922, 2019.

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

[10] 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. *JMLR (to appear), arXiv preprint arXiv:1809.04198*, 2019.

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

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

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

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

[15] G. Gidel, F. Pedregosa, and S. Lacoste-Julien. Frank-wolfe splitting via augmented lagrangian method. In *International Conference on Artificial Intelligence and Statistics*, pages 1456–1465, 2018.

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

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

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

[19] Q. Jiang, O. Adigun, H. Narasimhan, M. M. Fard, and M. Gupta. Optimizing black-box metrics with adaptive surrogates. *ArXiv:2002.08605*, 2020.

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

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

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

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

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

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

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

[27] D. Lewis. Evaluating text categorization. In *HLT Workshop on Speech and Natural Language*, 1991.

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

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

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

[31] H. Narasimhan, A. Cotter, Y. Zhou, S. Wang, and W. Guo. Approximate heavily-constrained learning with Lagrange multiplier models. In *NeurIPS*, 2020, to appear.

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

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

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

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

[36] S. Sabato, S. Ben-David, and S. Shalev-Shwartz. Multiclass learnability and the erm principle. In *Conference on Learning Theory*, 2003.

[37] 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.

[38] C. Scott. Surrogate losses and regret bounds for cost-sensitive classification with example-dependent costs. In *International Conference on Machine Learning*, 2011.

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

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

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

[42] 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.

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

[44] B. Yan, O. Koyejo, K. Zhong, and P. Ravikumar. Binary classification with karmic, threshold-quasi-concave metrics. In *ICML*, 2018.

[45] B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. In *International Conference on Data Mining*, 2003.

[46] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi. Fairness constraints: Mechanisms for fair classification. In *AISTATS*, 2017.

A Proofs

Before we give the proofs, we define some terms necessary for the proofs and make the assumptions on the problem more explicit.

A.1 Proof Setup

A.1.1 Problem Assumptions

We had made several assumptions on the problem in the paper, which we recall here for reference.

1. The sufficient statistics functions $\sigma_1, \ldots, \sigma_d$ are bounded between 0 and 1.
2. The functions ψ, ϕ_k are convex.
3. The function $\psi : [0,1]^d \to \mathbb{R}$ is also bounded between 0 and R
4. The functions ψ and ϕ_k are L-Lipschitz, i.e. $|\psi(\mathbf{u}) - \psi(\mathbf{u}')| \leq L||\mathbf{u} - \mathbf{u}'||_2$ and $|\phi_k(\mathbf{u}) - \phi_k(\mathbf{u}')| \leq L||\mathbf{u} - \mathbf{u}'||_2$
5. The function ψ is β-smooth, i.e. $||\nabla\psi(\mathbf{u}) - \nabla\psi(\mathbf{u}')||_2 \leq \beta||\mathbf{u} - \mathbf{u}'||_2$
6. The sets $\mathcal{C} \subseteq [0,1]^d$ and $\mathcal{F} \subseteq \mathbb{R}^d$ are full-dimensional, i.e. interiors are not empty.
7. The interiors of the sets $\mathcal{C}$ and $\mathcal{F}$ intersect. For some $r > 0$, there exists $\mathbf{c} \in \mathcal{C} \cap \mathcal{F}$ such that $B(\mathbf{c}, r) \subseteq \mathcal{C} \cap \mathcal{F}$.

The last two assumptions are made only for convenience and can be relaxed to Assumption 1. Any problem for which $\mathcal{C}$ and $\mathcal{F}$ are not full dimensional, can be converted to an equivalent problem where they are full dimensional by projecting the sufficient statistic functions σ on to an appropriate affine space. Note that as the sufficient statistic functions take values $[0,1]$, the set $\mathcal{C}$ is always a subset of $[0,1]^d$, and we can also assume without loss of generality $\mathcal{F} \subseteq [0,1]^d$.

In the proofs of the Theorems below, we will use c to denote constants independent of the LMO error ρ and number of iterations T. The value of c can change even in consecutive expressions, to avoid cluttering the proof with unnecessary subscripts.

A.1.2 Extra Definitions

Definition 1 (Linear Minimization Oracle). *Let $\rho, \rho', \delta \in (0,1)$. A linear minimization oracle, denoted by Ω, takes a loss vector $\mathbf{a} \in \mathbb{R}^d$ and a sample S as input, and outputs a classifier $\widehat{g}$ and an estimate of its confusion vector $\widetilde{\mathbf{u}} \in \mathbb{R}^d$. We say the Ω is a (ρ, ρ', δ)-approximate LMO for sample size N if for all $\mathbf{a} \in \mathbb{R}^d$, it outputs $(\widehat{g}, \widetilde{\mathbf{u}}) = \Omega(\mathbf{a}; S)$ such that:*

$$
\begin{aligned}
\langle \mathbf{a}, \mathbf{C}[\widehat{g}] \rangle &\leq \min_{h:\mathcal{X}\to\Delta_n} \langle \mathbf{a}, \mathbf{C}[h] \rangle + \rho' \|\mathbf{a}\| \\
\|\mathbf{C}[\widehat{g}] - \widetilde{\mathbf{u}}\| &\leq \rho.
\end{aligned}
$$

where the second inequality above is only required to hold with probability $1 - \delta$ over the sample S. The approximation constants ρ and ρ' may in turn depend on the sample size N, the dimension d and the confidence level δ.

Definition 2 (Fat Achievable Set). *The set $\mathcal{C}_\rho$ is defined as follows:*

$$\mathcal{C}_\rho = \mathcal{C} + B(\mathbf{0}, \rho) = \{\mathbf{u} + \mathbf{r} : \mathbf{u} \in \mathcal{C}, \mathbf{r} \in B(\mathbf{0}, \rho)\}$$

Definition 3 (Augmented Lagrangian). *The Augmented Lagrangian $\mathcal{L} : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is defined as*

$$\mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \psi(\mathbf{u}) + \psi(\mathbf{v}) + \frac{\lambda}{2}\|\mathbf{u} - \mathbf{v}\|^2 + \mathbf{w}^\top(\mathbf{u} - \mathbf{v})$$

Simple algebra shows that $\mathcal{L}(.,.,\mathbf{w})$ is convex, Lipschitz continuous and smooth. We will require the following related inequalities for our Theorems.

Proposition 4. *For all $\mathbf{w} \in \mathbb{R}^d$, we have*

$$
\begin{aligned}
|\psi(\mathbf{u}) + \psi(\mathbf{v}) - \psi(\mathbf{u}') - \psi(\mathbf{v}')| &\leq 2L\sqrt{\|\mathbf{u} - \mathbf{u}'\|^2 + \|\mathbf{v} - \mathbf{v}'\|^2} \\
\|\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{w}) - \nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}', \mathbf{v}', \mathbf{w})\| &\leq \beta_\lambda \|[\mathbf{u} - \mathbf{u}', \mathbf{v} - \mathbf{v}']\| \\
\|\nabla_{\mathbf{v}}\mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{w}) - \nabla_{\mathbf{v}}\mathcal{L}(\mathbf{u}', \mathbf{v}', \mathbf{w})\| &\leq \beta_\lambda \|[\mathbf{u} - \mathbf{u}', \mathbf{v} - \mathbf{v}']\|
\end{aligned}
$$

where we use $\nabla_{\mathbf{u}}$ and $\nabla_{\mathbf{v}}$ to denote the gradient w.r.t. the first and second arguments of $\mathcal{L}$, and $\beta_\lambda = 2\beta + 2\lambda$.

Definition 4 (Dual Function)**.** *The dual function* $\xi : \mathbb{R}^d \to \mathbb{R}$ *is defined as*

$$\xi(\mathbf{w}) = \min_{\mathbf{u}\in\mathcal{C}_\rho, \mathbf{v}\in\mathcal{F}} \mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{w})$$

We also use $\widehat{\mathbf{u}}(\mathbf{w}), \widehat{\mathbf{v}}(\mathbf{w})$ to denote any arbitrary minimizer of $\mathcal{L}(.,.,\mathbf{w})$ over $\mathcal{C}_\rho \times \mathcal{F}$. Thus $\xi(\mathbf{w}) = \mathcal{L}(\widehat{\mathbf{u}}(\mathbf{w}), \widehat{\mathbf{v}}(\mathbf{w}), \mathbf{w})$.

Let the maximum value of the dual function be ξ^*. By the min-max Theorem we have that

$$\xi^* = \max_{\mathbf{w}\in\mathbb{R}^d} \min_{\mathbf{u}\in\mathcal{C}_\rho, \mathbf{v}\in\mathcal{F}} \mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \min_{\mathbf{u}\in\mathcal{C}_\rho, \mathbf{v}\in\mathcal{F}} \max_{\mathbf{w}\in\mathbb{R}^d} \mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \min_{\mathbf{u}\in\mathcal{C}_\rho\cap\mathcal{F}} 2\psi(\mathbf{u})$$

The last equality follows from the observation that if $\mathbf{u} \neq \mathbf{v}$ then $\max_{\mathbf{w}\in\mathbb{R}^d} \mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \infty$.

Let $\mathbf{u}^* \in \mathcal{C}_\rho \cap \mathcal{F}$ such that

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

Let $\mathcal{W}^* = \mathrm{argmax}_{\mathbf{w}\in\mathbb{R}^d} \xi(\mathbf{w}) \subseteq \mathbb{R}^d$.

Definition 5 (Primal and Dual gaps)**.** *For any* $\mathbf{u} \in \mathcal{C}_\rho, \mathbf{v} \in \mathcal{F}$ *and* $\mathbf{w} \in \mathbb{R}^d$*, we define the primal and dual gaps as follows:*

$$\Delta^{(p)}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{w}) - \min_{\mathbf{u}\in\mathcal{C}_\rho, \mathbf{v}\in\mathcal{F}} \mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{w}) - \xi(\mathbf{w})$$
$$\Delta^{(d)}(\mathbf{w}) = \xi^* - \xi(\mathbf{w}) = 2\psi(\mathbf{u}^*) - \xi(\mathbf{w})$$

and define the total gap as $\Delta(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \Delta^{(p)}(\mathbf{u}, \mathbf{v}, \mathbf{w}) + \Delta^{(d)}(\mathbf{w})$.

In the Theorems and Lemmas below, we will refer to the iterates $\mathbf{u}_t, \mathbf{v}_t, \widetilde{\mathbf{u}}_t, \widetilde{\mathbf{v}}_t$ in the Algorithm 1. 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, $(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_t)$.

The overloading of notation for ρ in the definition of the duality gaps and the LMO confusion vector estimation error is intentional. In our analysis of the algorithm using the duality gap, we will set ρ to be exactly equal to the confusion vector estimation error in the `plug-in` algorithm referred to by Algorithm 1.

We will require the use of Theorem 1 and Corollary 1 from Gidel et al. [15], which we restate here in our notation. We use the following facts to transform their Theorem.

$$|\psi(\mathbf{u}) + \psi(\mathbf{v}) - \psi(\mathbf{u}') - \psi(\mathbf{v}')| \leq 2L\|[\mathbf{u} - \mathbf{u}', \mathbf{v} - \mathbf{v}']\|$$
$$\|[I, -I]^\top [-I, I]\| = 2$$
$$(\mathrm{diam}(F))^2 \leq d$$
$$(\mathrm{diam}(\mathcal{C}_\rho))^2 \leq 2d + 2\rho^2$$
$$(\mathrm{diam}(\mathcal{C}_\rho \times \mathcal{F}))^2 \leq 3d + 2\rho^2$$

where $\|M\|$ of a matrix M refers to its spectral norm, and 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 $\mathcal{A}$. We will use ζ^2 as a shorthand for $3d + 2\rho^2$.

Theorem. *There exists a constant* $\alpha > 0$ *such that*

$$\xi^* - \xi(\mathbf{w}) \geq \frac{1}{2L_\lambda\zeta^2} \min\left\{\alpha^2 \mathrm{dist}(\mathbf{w}, \mathcal{W}^*)^2, \alpha L_\lambda \zeta^2 \mathrm{dist}(\mathbf{w}, \mathcal{W}^*)\right\}$$
$$||\nabla\xi(\mathbf{w})|| \geq \frac{1}{2L_\lambda\zeta^2} \min\left\{\alpha^2 \mathrm{dist}(\mathbf{w}, \mathcal{W}^*), \alpha L_\lambda \zeta^2\right\}$$
$$||\nabla\xi(\mathbf{w})|| \geq \frac{\alpha}{\sqrt{2L_\lambda\zeta^2}} \min\left\{\sqrt{\xi^* - \xi(\mathbf{w})}, \sqrt{\frac{L_\lambda\zeta^2}{2}}\right\}$$

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

A.2 Proof of Proposition 2

Proposition (LMO$_\mathcal{C}$ through Bayes-optimal Classifier). *Suppose we wish to minimize* $\langle \mathbf{a}, \mathbf{u} \rangle$ *over* $\mathbf{u} \in \mathcal{C}$*. Define the example-dependent loss matrix* $\mathbf{L} : \mathcal{X} \to \mathbb{R}^{n\times n}$ *as* $L_{j,k}(x) = \sum_{i=1}^d a_i \sigma_i(x, j, k)$*. Then the solution to* *the linear minimization problem is directly given by the Bayes-optimal classifier for this loss matrix. Specifically, construct a classifier g^* : $\mathcal{X} \to [n]$ with*

$$g^*(x) = \operatorname*{argmin}_{\widehat{y}\in[n]} \sum_{j=1}^{n} \eta_j(x)\, L_{j,\widehat{y}}(x),$$

where $\eta_j(x) = \mathbf{P}(Y = 1|x)$ is the class-conditional probability. Then $C[g^*] \in \operatorname{argmin}_{\mathbf{u}\in\mathcal{C}} \langle \mathbf{a}, \mathbf{u}\rangle$.

Proof.

$$\begin{aligned}
\min_{\mathbf{u}\in\mathcal{C}} \langle \mathbf{a}, \mathbf{u}\rangle &= \min_{g\in\mathcal{H}} \sum_{i=1}^{d} a_i \mathbf{E}_{X\sim\mu}\left[\mathbf{E}_{Y\sim\eta(X)}[\mathbf{E}_{\widehat{Y}\sim g(X)}[\sigma_i(X,Y,\widehat{Y})]]\right] \\
&= \mathbf{E}_{X\sim\mu}\left[\min_{\mathbf{g}\in\Delta_n} \mathbf{E}_{\widehat{Y}\sim\mathbf{g}}\left[\mathbf{E}_{Y\sim\eta(X)}\left[\sum_{i=1}^{d} a_i\sigma_i(X,Y,\widehat{Y})\right]\right]\right] \\
&= \mathbf{E}_{X\sim\mu}\left[\min_{\widehat{y}\in[n]} \sum_{j=1}^{n} \eta_j(X) \sum_{i=1}^{d} a_i\sigma_i(X,j,\widehat{y})\right] \\
&= \mathbf{E}_{X\sim\mu}\left[\min_{\widehat{y}\in[n]} \sum_{j=1}^{n} \eta_j(X) L_{j,\widehat{y}}(X)\right]
\end{aligned}$$

Now,

$$\begin{aligned}
\langle \mathbf{a}, C[g^*]\rangle &= \sum_{i=1}^{d} a_i \mathbf{E}_{X\sim\mu}\left[\mathbf{E}_{Y\sim\eta(X)}\left[\mathbf{E}_{\widehat{Y}\sim g^*(X)}\left[\sigma_i(X,Y,\widehat{Y})\right]\right]\right] \\
&= \mathbf{E}_{X\sim\mu}\left[\mathbf{E}_{Y\sim\eta(X)}\left[\sum_{i=1}^{d} a_i\sigma_i(X,Y,g^*(X))\right]\right] \\
&= \mathbf{E}_{X\sim\mu}\left[\sum_{j=1}^{n} \eta_j(X) \sum_{i=1}^{d} a_i\sigma_i(X,j,g^*(X))\right] \\
&= \mathbf{E}_{X\sim\mu}\left[\sum_{j=1}^{n} \eta_j(X) L_{j,g^*(X)}(X)\right] \\
&= \mathbf{E}_{X\sim\mu}\left[\min_{\widehat{y}\in[n]} \sum_{j=1}^{n} \eta_j(X) L_{j,\widehat{y}}(X)\right]
\end{aligned}$$

where the last equation follows from construction of g^*. □

A.3 Proof of Theorem 3

Theorem. *Let h^* denote the optimal feasible solution for* (OP1)*, i.e. $\phi_k(C[h^*]) \le 0, \forall k$ and $\psi(C[h^*]) \le \psi(C[h])$ for all h that is feasible. Under the regularity assumptions, for large enough λ and an appropriate step-size parameter C, there exists an $\bar{\epsilon} > 0$ such that, for all $\epsilon \le \bar{\epsilon}$, and $T \ge \frac{C}{\epsilon^2}$, with probability $1 - \delta$ over draw of the training samples S i.i.d. from D, the classifier $\widehat{h}$ returned by Algorithm 1 is near-optimal and near-feasible:*

$$\begin{aligned}
\textbf{\textit{Optimality}}: \;& \psi(C[\widehat{h}]) \le \psi(C[h^*]) + c\sqrt{\omega} + \epsilon, \\
\textbf{\textit{Feasibility}}: \;& \phi_k(C[\widehat{h}]) \le c\sqrt{\omega} + L\epsilon, \; \forall k \in [K],
\end{aligned}$$

where $\omega = \sqrt{d}\mathbf{E}||\eta(X) - \widehat{\eta}(X)||_1 + d\sqrt{\frac{d\log(d)+\log(Nn^2)+\log(1/\delta)}{N}}$ captures the approximation level of the LMO given by Algorithm 2, and $c > 0$ is a constant not dependent on the number of iterations T and the training samples.

Proof. Firstly, we prove in Corollary 7 that the Algorithm 2 gives an approximate LMO over $\mathcal{C}$ even though it uses only finite data. These Lemmas are more general than those in Narasimhan et al. (2015) [33], because they accommodate more general sufficient statistics functions σ.

Secondly, we show that the usage of an approximate LMO in Equations (4), and (5) does not affect the convergence results by Gidel et al. [15]. They measure the sub-optimality of an iterate using a duality gap measure. In Lemma 9 we show that a similar bound on the duality gap can be derived with an approximate LMO over $\mathcal{C}$ as well.

Thirdly, we use the strict feasibilty assumption to convert a bound on the duality gap into a bound on the sub-optimality of problem (OP2) in Lemma 8.

Lemmas 9 can be applied to Lemma 8 setting both τ and κ to be equal to $\frac{c}{T} + c(\rho + \rho')$. In both the inequalities, the $\sqrt{\kappa}$ term dominates, and hence

$$||C[h_b] - \mathbf{v}_b||_2^2 \leq c(\rho + \rho') + \frac{c}{T}$$
$$\psi(C[h_b]) \leq \min_{\mathbf{u} \in \mathcal{C} \cap \mathcal{F}} \psi(\mathbf{u}) + c\sqrt{\rho + \rho' + \frac{1}{T}}$$

For large enough T, these can be simplified as follows,

$$||C[h_b] - \mathbf{v}_b|| \leq c\sqrt{\rho + \rho'} + \frac{c}{\sqrt{T}}$$
$$\psi(C[h_b]) \leq \min_{\mathbf{u} \in \mathcal{C} \cap \mathcal{F}} \psi(\mathbf{u}) + c\sqrt{\rho + \rho'} + \frac{c}{\sqrt{T}}$$

Observing that $\mathbf{v}_b \in \mathcal{F}$ and the constraint functions ϕ_k are L-Lipschitz, we get the Theorem statement. The expressions for $\omega = \rho + \rho'$ follow from Corollary 7. □

A.3.1 LMO Lemmas

Lemma 5. *Let* $\mathbf{a} \in \mathbb{R}^d$. *Let* $\widehat{g}, \widetilde{\mathbf{u}} = \texttt{plug-in}(\mathbf{a})$ *as in Algorithm 2, then*

$$\langle \mathbf{a}, C[\widehat{g}] \rangle \leq \min_{\mathbf{u} \in \mathcal{C}} \langle \mathbf{a}, \mathbf{u} \rangle + 2||\mathbf{a}||_2 \sqrt{d} \mathbf{E} ||\eta(X) - \widehat{\eta}(X)||_1$$

for some constant $c_3 > 0$.

Proof. Fix some $\mathbf{a} \in \mathbb{R}^d$. Let $\mathbf{L} : \mathcal{X} {\rightarrow} \mathbb{R}^{n \times n}$, be such that,

$$L_{j,k}(x) = \sum_{i=1}^{d} a_i \sigma_i(x, j, k) \leq ||\mathbf{a}||_1 \leq \sqrt{d} ||\mathbf{a}||_2$$

From Proposition 2, the Bayes optimal classifier $g^* : \mathcal{X} {\rightarrow} [n]$ is

$$g^*(x) = \operatorname*{argmin}_{\widehat{y} \in [n]} \sum_{j=1}^{n} \eta_j(x) \, L_{j, \widehat{y}}(x),$$

Recall that $\widehat{g}$ is the same as g^* above, with η replaced by $\widehat{\eta}$. We have that,

$$\begin{aligned}
\langle \mathbf{a}, C[\widehat{g}] \rangle &= \mathbf{E}_X [\mathbf{E}_{Y \sim \eta(X)} [\sum_{i=1}^{d} a_i \sigma_i(X, Y, \widehat{g}(X))]] \\
&= \mathbf{E}_X \left[\sum_{y=1}^{n} \eta_y(X) L_{y, \widehat{g}(X)}(X) \right] \\
&= \mathbf{E}_X \left[\sum_{y=1}^{n} (\eta_y(X) - \widehat{\eta}_y(X)) L_{y, \widehat{g}(X)}(X) \right] + \mathbf{E}_X \left[\sum_{y=1}^{n} \widehat{\eta}_y(X) L_{y, \widehat{g}(X)}(X) \right] \\
&\leq ||\mathbf{a}||_2 \sqrt{d} \mathbf{E}_X [||\eta(X) - \widehat{\eta}(X)||_1] + \mathbf{E}_X \left[\sum_{y=1}^{n} \widehat{\eta}_y(X) L_{y, \widehat{g}(X)}(X) \right] \\
&\leq ||\mathbf{a}||_2 \sqrt{d} \mathbf{E}_X [||\eta(X) - \widehat{\eta}(X)||_1] + \mathbf{E}_X \left[\sum_{y=1}^{n} \widehat{\eta}_y(X) L_{y, g^*(X)}(X) \right] \\
&\leq ||\mathbf{a}||_2 \sqrt{d} \mathbf{E}_X [||\eta(X) - \widehat{\eta}(X)||_1] + \mathbf{E}_X \left[\sum_{y=1}^{n} (\widehat{\eta}_y(X) - \eta_y(X) L_{y, g^*(X)}(X) \right] + \mathbf{E}_X \left[\sum_{y=1}^{n} (\eta_y(X) L_{y, g^*(X)}(X) \right] \\
&\leq 2||\mathbf{a}||_2 \sqrt{d} \mathbf{E}_X \left[||\eta(X) - \widehat{\eta}(X)||_1 \right] + \min_{\mathbf{u} \in \mathbf{C}} \langle \mathbf{a}, \mathbf{u} \rangle
\end{aligned}$$

□

Lemma 6. *Let $\widehat{g}^{\mathbf{a}}, \widetilde{\mathbf{u}}^{\mathbf{a}} =$ `plug-in`$(\mathbf{a})$ as in Algorithm 2, then with probability $1-\delta$ over the samples $\{(x_{N/2}, y_{N/2}), \ldots, (x_N, y_N)\}$, we have that for all $\mathbf{a} \in \mathbb{R}^d$*

$$||C[\widehat{g}^{\mathbf{a}}] - \widetilde{\mathbf{u}}^{\mathbf{a}}||_2 \leq cd\sqrt{\frac{d\log(d) + \log(Nn^2) + \log(1/\delta)}{N}}$$

where c is an absolute constant.

Proof. Fix some $z \in [d]$. We have for any $\mathbf{a} \in \mathbb{R}^d$,

$$C_z[\widehat{g}^{\mathbf{a}}] = \mathbf{E}_{X,Y}\sigma_z(X, Y, \widehat{g}^{\mathbf{a}}(X)) = \mathbf{E}_D\left[\sigma_z(X, Y, \widehat{g}^{\mathbf{a}}(X))\right]$$
$$\widetilde{u}^{\mathbf{a}}_z = \frac{2}{N}\sum_{j=N/2}^{N}\sigma_z(x_j, y_j, \widehat{g}^{\mathbf{a}}(x_j)) = \mathbf{E}_S\left[\sigma_z(X, Y, \widehat{g}^{\mathbf{a}}(X))\right]$$

where we abuse notation by denoting the empirical expectation over the last $N/2$ samples as $\mathbf{E}_S$ and the population expectation as $\mathbf{E}_D$.

For any $x \in \mathcal{X}, \widehat{y} \in [n]$, let $\theta(x, \widehat{y}) \in \mathbb{R}^d$ be such that

$$\theta_i(x, \widehat{y}) = \sum_{y=1}^{n}\widehat{\eta}_y(x)\sigma_i(x, y, \widehat{y}).$$

Then, by definition of $\widehat{g}^{\mathbf{a}}$, we have that

$$\widehat{g}^{\mathbf{a}}(x) = \operatorname{argmin}_{\widehat{y}\in[n]}\mathbf{a}^\top\boldsymbol{\theta}(x, \widehat{y}).$$

We have that the Natarajan dimension d_{Nat} of the function class

$$\mathcal{G} = \{\widehat{g}^{\mathbf{a}}(x) = \operatorname{argmin}_{\widehat{y}\in[n]}\mathbf{a}^\top\theta(x, \widehat{y}) : \mathbf{a} \in \mathbb{R}^d\} \subseteq [n]^{\mathcal{X}}$$

is $O(d\log(d))$ [36]. The growth function $\Pi_{\mathcal{G}}(N)$ denoting the number of distinct labellings of N points is given by Cesa-Bianchi and Haussler [6] as,

$$\Pi_{\mathcal{G}}(N) \leq d_{\text{Nat}}N^{d_{\text{Nat}}}n^{2d_{\text{Nat}}}.$$

Using standard Hoeffding inequality and uniform convergence arguments, we have that with probability $1-\delta$

$$\begin{aligned}\sup_{\mathbf{a}\in\mathbb{R}^d}|C_z[\widehat{g}^{\mathbf{a}}] - \widetilde{u}^{\mathbf{a}}_z| &= \sup_{g\in\mathcal{G}}\left|\mathbf{E}_S\left[\sigma_z(X, Y, g(X))\right] - \mathbf{E}_D\left[\sigma_z(X, Y, g(X))\right]\right| \\ &\leq c\left(\sqrt{\frac{\log(\Pi_{\mathcal{G}}(N)) + \log(1/\delta)}{N}}\right) \\ &\leq c\left(\sqrt{\frac{d\log(d) + \log(Nn^2) + \log(1/\delta)}{N}}\right)\end{aligned}$$

We thus have that,

$$\begin{aligned}\sup_{\mathbf{a}\in\mathbb{R}^d}||C[\widehat{g}^{\mathbf{a}}] - \widetilde{\mathbf{u}}^{\mathbf{a}}||_2 &\leq \sup_{\mathbf{a}\in\mathbb{R}^d}||C[\widehat{g}^{\mathbf{a}}] - \widetilde{\mathbf{u}}^{\mathbf{a}}||_1 \\ &= \sup_{\mathbf{a}\in\mathbb{R}^d}\sum_{z=1}^{d}|C_z[\widehat{g}^{\mathbf{a}}] - \widetilde{u}^{\mathbf{a}}_z| \\ &\leq \sum_{z=1}^{d}\sup_{\mathbf{a}\in\mathbb{R}^d}|C_z[\widehat{g}^{\mathbf{a}}] - \widetilde{u}^{\mathbf{a}}_z| \\ &\leq c\left(d\sqrt{\frac{d\log(d) + \log(Nn^2) + \log(1/\delta)}{N}}\right).\end{aligned}$$

where the last statement holds with probability $1-\delta$. □

Corollary 7. *The function* `plug-in` *in Algorithm 2 is a (ρ, ρ', δ)-approximate* LMO *with $\rho = cd\sqrt{\frac{d\log(d)+\log(Nn^2)+\log(1/\delta)}{N}}$ and $\rho' = 2\sqrt{d}\mathbf{E}||\eta(X) - \widehat{\eta}(X)||_1$ for some constant $c > 0$.*

We will fix a δ probability of failure throughout the rest of the proof, and assume that the training sample S is "good", in which case the empirical confusion vector output by the `plug-in` algorithm is ρ close to the true confusion vector of the classifier whenever it is called by Algorithm 1.

A.3.2 Converting Duality Gap Bounds to Primal Sub-Optimality Bounds

Lemma 8. *Let $\mathbf{g} : \mathcal{X} \to \Delta_n$ be a randomized classifier, and $\mathbf{u} \in \mathbb{R}^d$ be such that $\|\mathbf{u} - C[\mathbf{g}]\| \leq \rho$. Let $\mathbf{v} \in \mathcal{F}, \mathbf{w} \in \mathbb{R}^d$ be such that $\Delta(\mathbf{u}, \mathbf{v}, \mathbf{w}) \leq \tau$ and $||\mathbf{u} - \mathbf{v}||^2 \leq \kappa$. Then,*

$$\psi(C[\mathbf{g}]) \leq \min_{\mathbf{u}' \in \mathcal{C} \cap \mathcal{F}} \psi(\mathbf{u}') + c\tau + c\sqrt{\kappa} + L\rho$$

$$\|C[\mathbf{g}] - \mathbf{v}\| \leq \rho + \sqrt{\kappa}$$

for some constant $c > 0$.

Proof. The second inequality in the lemma trivially follows from the triangle inequality. We will prove the first inequality below.

By construction, $\mathbf{u} \in \mathcal{C}_\rho$. As $\Delta(\mathbf{u}, \mathbf{v}, \mathbf{w}) \leq \tau$ we have,

$$\Delta^{(\mathrm{p})}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{w}) - \min_{\mathbf{u}' \in \mathcal{C}_\rho, \mathbf{v}' \in \mathcal{F}} \mathcal{L}(\mathbf{u}', \mathbf{v}', \mathbf{w}) \leq \tau \tag{6}$$

$$\Delta^{(\mathrm{d})}(\mathbf{w}) = 2\psi(\mathbf{u}^*) - \min_{\mathbf{u}' \in \mathcal{C}_\rho, \mathbf{v}' \in \mathcal{F}} \mathcal{L}(\mathbf{u}', \mathbf{v}', \mathbf{w}) \leq \tau \tag{7}$$

where $\mathbf{u}^* \in \operatorname{argmin}_{\mathbf{u}' \in \mathcal{C}_\rho \cap \mathcal{F}} \psi(\mathbf{u}')$. Setting $\mathbf{u}' = \mathbf{v}' = \mathbf{u}^*$ in the second term of Eqn. (6), we get

$$\psi(\mathbf{u}) + \psi(\mathbf{v}) + \mathbf{w}^T(\mathbf{u} - \mathbf{v}) + \frac{\lambda}{2}\|\mathbf{u} - \mathbf{v}\|^2 \leq 2\psi(\mathbf{u}^*) + \tau\,. \tag{8}$$

Now from our assumption that there exists a ball of radius r contained in $\mathcal{C} \cap \mathcal{F}$, we can set $\mathbf{u}', \mathbf{v}' = \mathbf{c} \pm \frac{r}{||\mathbf{w}||}\mathbf{w}$ in the second term of Eqn. (7) to get

$$2\psi(\mathbf{u}^*) \leq \psi(\mathbf{u}') + \psi(\mathbf{v}') - 2r\|\mathbf{w}\| + 2\lambda r^2 + \tau. \tag{9}$$

This can be reduced to a bound on $||\widehat{\mathbf{w}}||$,

$$\|\mathbf{w}\| \leq \frac{2R}{r} + 2\lambda r + \frac{\tau}{r} \tag{10}$$

Eqn. (8) becomes the following by Cauchy-Schwarz:

$$\psi(\mathbf{u}) + \psi(\mathbf{v}) \leq 2\psi(\mathbf{u}^*) + \tau - \mathbf{w}^\top(\mathbf{u} - \mathbf{v}) - \frac{\lambda}{2}\|\mathbf{u} - \mathbf{v}\|^2 \leq 2\psi(\mathbf{u}^*) + \tau + \left(\frac{2R}{r} + 2\lambda r + \frac{\tau}{r}\right)\sqrt{\kappa}. \tag{11}$$

As ψ is L-Lipschitz, we have

$$\psi(\mathbf{u}) - \psi(\mathbf{v}) \leq L\|\mathbf{u} - \mathbf{v}\| \leq L\sqrt{\kappa} \tag{12}$$

Adding Eqns. (11) and 12 and dividing by 2, we get

$$\psi(\mathbf{u}) \leq \min_{\mathbf{u}' \in \mathcal{C}_\rho \cap \mathcal{F}} \psi(\mathbf{u}') + \frac{\tau}{2} + \frac{\left(\frac{2R}{r} + 2\lambda r + \frac{\tau}{r}\right) + L}{2}\sqrt{\kappa}$$

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

$$\begin{aligned}
\psi(C[\mathbf{g}]) &\leq \psi(\mathbf{u}) + L\|\mathbf{u} - C[\mathbf{g}]\| \\
&\leq \min_{\mathbf{u}' \in \mathcal{C}_\rho \cap \mathcal{F}} \psi(\mathbf{u}') + \frac{\tau}{2} + \frac{\left(\frac{2R}{r} + 2\lambda r + \frac{\tau}{r}\right) + L}{2}\sqrt{\kappa} + L\rho \\
&\leq \min_{\mathbf{u}' \in \mathcal{C} \cap \mathcal{F}} \psi(\mathbf{u}') + \frac{\tau}{2} + \frac{\left(\frac{2R}{r} + 2\lambda r + \frac{\tau}{r}\right) + L}{2}\sqrt{\kappa} + L\rho
\end{aligned}$$

□

A.3.3 Bounding the Duality Gap

Lemma 9. *Let $b \in [T]$ be such that $\widehat{h} = h_b$ in Algorithm 1. Let the* `plug-in` *sub-routine used be a (ρ, ρ', δ)-approximate LMO. For large enough T and λ, with probability $1 - \delta$ over the training samples we have that*

$$\Delta(\mathbf{u}_b, \mathbf{v}_b, \mathbf{w}_{b-1}) \leq c(\rho + \rho') + \frac{c}{T}$$

$$||\mathbf{u}_b - \mathbf{v}_b||^2 \leq c(\rho + \rho') + \frac{c}{T}$$

where $h_b, \mathbf{v}_b, \mathbf{w}_{b-1}$ are as defined in Algorithm 1, and c is a constant independent of ρ, ρ' and T.

Proof. For large λ and T, the conditions in Corollary 16 and Lemma 17 are satisfied, and hence the Lemma follows directly. □

Lemma 10. *For all* $\mathbf{u} \in \mathcal{C}_\rho$, $\mathbf{v} \in \mathcal{F}$ *and* $\mathbf{w} \in \mathbb{R}^d$

$$\|(\mathbf{u}-\mathbf{v}) - (\widehat{\mathbf{u}}(\mathbf{w}) - \widehat{\mathbf{v}}(\mathbf{w}))\|^2 \leq \frac{2}{\lambda}\left(\mathcal{L}(\mathbf{u},\mathbf{v},\mathbf{w}) - \mathcal{L}(\widehat{\mathbf{u}}(\mathbf{w}), \widehat{\mathbf{v}}(\mathbf{w}), \mathbf{w})\right) \tag{13}$$

where $\widehat{\mathbf{u}}(\mathbf{w}), \widehat{\mathbf{v}}(\mathbf{w}) \in \operatorname{argmin}_{\mathbf{u}\in\mathcal{C}_\rho, \mathbf{v}\in\mathcal{F}} \mathcal{L}(\mathbf{u},\mathbf{v},\mathbf{w})$ *are functions of* $\mathbf{w}$.

Proof. We drop the dependence on $\mathbf{w}$ in $\widehat{\mathbf{u}}, \widehat{\mathbf{v}}$ for simplicity below.

By convexity of ψ we have that,

$$\psi(\mathbf{u}) - \psi(\widehat{\mathbf{u}}) \geq (\nabla\psi(\widehat{\mathbf{u}}))^T (\mathbf{u} - \widehat{\mathbf{u}}) \text{ and } \psi(\mathbf{v}) - \psi(\widehat{\mathbf{v}}) \geq (\nabla\psi(\widehat{\mathbf{v}}))^T (\mathbf{v} - \widehat{\mathbf{v}})$$

then by simple algebra,

$$\begin{aligned}
&\mathcal{L}(\mathbf{u},\mathbf{v},\mathbf{w}) - \mathcal{L}(\widehat{\mathbf{u}},\widehat{\mathbf{v}},\mathbf{w}) \\
&= \psi(\mathbf{u}) - \psi(\widehat{\mathbf{u}}) + \psi(\mathbf{v}) - \psi(\widehat{\mathbf{v}}) + \mathbf{w}^\top(\mathbf{u} - \mathbf{v} - \widehat{\mathbf{u}} + \widehat{\mathbf{v}}) + \frac{\lambda}{2}(\|\mathbf{u}-\mathbf{v}\|^2 - \|\widehat{\mathbf{u}} - \widehat{\mathbf{v}}\|^2) \\
&\geq (\nabla\psi(\widehat{\mathbf{u}}) + \mathbf{w})^\top (\mathbf{u} - \widehat{\mathbf{u}}) + (\nabla\psi(\widehat{\mathbf{v}}) - \mathbf{w})^\top (\mathbf{v} - \widehat{\mathbf{v}}) + \frac{\lambda}{2}(\|\mathbf{u}-\mathbf{v}\|^2 - \|\widehat{\mathbf{u}} - \widehat{\mathbf{v}}\|^2) \\
&= (\nabla\psi(\widehat{\mathbf{u}}) + \mathbf{w} + \lambda(\widehat{\mathbf{u}} - \widehat{\mathbf{v}}))^\top (\mathbf{u} - \widehat{\mathbf{u}}) + (\nabla\psi(\widehat{\mathbf{v}}) - \mathbf{w} - \lambda(\widehat{\mathbf{u}} - \widehat{\mathbf{v}}))^\top (\mathbf{v} - \widehat{\mathbf{v}}) \\
&\quad + \frac{\lambda}{2}\|(\mathbf{u}-\mathbf{v}) - (\widehat{\mathbf{u}} - \widehat{\mathbf{v}})\|^2 \\
&= (\nabla_{\mathbf{u}}\mathcal{L}(\widehat{\mathbf{u}},\widehat{\mathbf{v}},\mathbf{w}))^\top(\mathbf{u} - \widehat{\mathbf{u}}) + (\nabla_{\mathbf{v}}\mathcal{L}(\widehat{\mathbf{u}},\widehat{\mathbf{v}},\mathbf{w}))^\top(\mathbf{v} - \widehat{\mathbf{v}}) + \frac{\lambda}{2}\|(\mathbf{u}-\mathbf{v}) - (\widehat{\mathbf{u}} - \widehat{\mathbf{v}})\|^2 \\
&\geq \frac{\lambda}{2}\|(\mathbf{u}-\mathbf{v}) - (\widehat{\mathbf{u}} - \widehat{\mathbf{v}})\|^2
\end{aligned}$$

The last inequality follows from the definition $\widehat{\mathbf{u}}, \widehat{\mathbf{v}}$. □

The next lemma captures the essence of what happens in one iteration of Algorithm 1 in lines 5-8. We use the same symbols as in the algorithm for ease of reference.

Lemma 11. *Let* $\mathbf{u}_{t-1} \in \mathcal{C}_\rho, \mathbf{v}_{t-1} \in \mathcal{F}, \mathbf{w}_{t-1} \in \mathbb{R}^d$. *Let* $\mathbf{a}_{t-1} = \nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}_{t-1}, \mathbf{v}_{t-1}, \mathbf{w}_{t-1})$ *and* $\mathbf{b}_{t-1} = \nabla_{\mathbf{v}}\mathcal{L}(\mathbf{u}_{t-1}, \mathbf{v}_{t-1}, \mathbf{w}_{t-1})$. *Let* Ω *be a* (ρ, ρ', δ)*-approximate LMO. Let* $\widehat{g}_t, \widetilde{\mathbf{u}}_t = \Omega(\mathbf{a}_{t-1}; S)$, *and* $\widetilde{\mathbf{v}}_t \in \operatorname{argmin}_{\mathbf{v}\in\mathcal{F}}\langle \mathbf{b}_{t-1}, \mathbf{v}\rangle$. *Let* $\mathbf{u}_t = (1-\gamma_t)\mathbf{u}_{t-1} + \gamma_t\widetilde{\mathbf{u}}_t$ *and* $\mathbf{v}_t = (1-\gamma_t)\mathbf{v}_{t-1} + \gamma_t\widetilde{\mathbf{v}}_t$. *Let* $\widehat{\mathbf{u}}_{t-1}, \widehat{\mathbf{v}}_{t-1} = \widehat{\mathbf{u}}(\mathbf{w}_{t-1}), \widehat{\mathbf{v}}(\mathbf{w}_{t-1})$ *as defined in Lemma 10. Then*

$$\begin{aligned}
&\mathcal{L}(\mathbf{u}_t, \mathbf{v}_t, \mathbf{w}_{t-1}) - \mathcal{L}(\widehat{\mathbf{u}}_{t-1}, \widehat{\mathbf{v}}_{t-1}, \mathbf{w}_{t-1}) \\
&\leq (1-\gamma_t)\left(\mathcal{L}(\mathbf{u}_{t-1}, \mathbf{v}_{t-1}, \mathbf{w}_{t-1}) - \mathcal{L}(\widehat{\mathbf{u}}_{t-1}, \widehat{\mathbf{v}}_{t-1}, \mathbf{w}_{t-1})\right) + \gamma_t\|\mathbf{a}_{t-1}\|(\rho + \rho') + \frac{1}{2}\beta_\lambda\gamma_t^2\zeta^2
\end{aligned}$$

Proof. Using smoothness,

$$\begin{aligned}
&\mathcal{L}(\mathbf{u}_t, \mathbf{v}_t, \mathbf{w}_{t-1}) - \mathcal{L}(\mathbf{u}_{t-1}, \mathbf{v}_{t-1}, \mathbf{w}_{t-1}) \\
&\leq \nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}_{t-1}, \mathbf{v}_{t-1}, \mathbf{w}_{t-1})^\top[\mathbf{u}_t - \mathbf{u}_{t-1}] + \nabla_{\mathbf{v}}\mathcal{L}(\mathbf{u}_{t-1}, \mathbf{v}_{t-1}, \mathbf{w}_{t-1})^\top[\mathbf{v}_t - \mathbf{v}_{t-1}] + \frac{\beta_\lambda}{2}\left(\|\mathbf{u}_t - \mathbf{u}_{t-1}\|^2 + \|\mathbf{v}_t - \mathbf{v}_{t-1}\|^2\right) \\
&= \mathbf{a}_{t-1}^\top[\gamma_t(\widetilde{\mathbf{u}}_t - \mathbf{u}_{t-1})] + \mathbf{b}_{t-1}^\top[\gamma_t(\widetilde{\mathbf{v}}_t - \mathbf{v}_{t-1})] + \frac{\beta_\lambda}{2}\gamma_t^2\left(\|\widetilde{\mathbf{u}}_t - \mathbf{u}_{t-1}\|^2\right) + \frac{\beta_\lambda}{2}\gamma^2\left(\|\widetilde{\mathbf{v}}_t - \mathbf{v}_{t-1}\|^2\right) \\
&\leq \gamma_t\mathbf{a}_{t-1}^\top(\widetilde{\mathbf{u}}_t - \mathbf{u}_{t-1}) + \gamma_t\mathbf{b}_{t-1}^\top(\widetilde{\mathbf{v}}_t - \mathbf{v}_{t-1}) + \gamma_t^2\frac{\beta_\lambda}{2}(\operatorname{diam}^2(\mathcal{C}_\rho) + \operatorname{diam}^2(\mathcal{F})) \\
&\leq \gamma_t\mathbf{a}_{t-1}^\top(\widetilde{\mathbf{u}}_t - \mathbf{u}_{t-1}) + \gamma_t\mathbf{b}_{t-1}^\top(\widehat{\mathbf{v}}_{t-1} - \mathbf{v}_{t-1}) + \frac{1}{2}\beta_\lambda\gamma_t^2\zeta^2 \\
&\leq \gamma_t\mathbf{a}_{t-1}^\top(\widehat{\mathbf{u}}_{t-1} - \mathbf{u}_{t-1}) + \gamma_t\|\mathbf{a}_{t-1}\|\rho' + \gamma_t\|\mathbf{a}_{t-1}\|\rho + \gamma_t\mathbf{b}_{t-1}^\top(\widetilde{\mathbf{v}}_t - \mathbf{v}_{t-1}) + \frac{1}{2}\beta_\lambda\gamma_t^2\zeta^2 \\
&\leq \gamma_t\left(\mathcal{L}(\widehat{\mathbf{u}}_{t-1}, \widehat{\mathbf{v}}_{t-1}, \mathbf{w}_{t-1}) - \mathcal{L}(\mathbf{u}_{t-1}, \mathbf{v}_{t-1}, \mathbf{w}_{t-1})\right) + \gamma_t\|\mathbf{a}_{t-1}\|(\rho + \rho') + \frac{1}{2}\beta_\lambda\gamma_t^2\zeta^2
\end{aligned}$$

Rearranging the terms we get the statement of the lemma. □

The next lemma captures the essence of what happens in one iteration of Algorithm 1 in Line 9. We use the same symbols as in the algorithm for ease of reference.

Lemma 12 (Variant of Fundamental Descent Lemma in Gidel et al. (2018) [15]). *Let $\mathbf{w}_t \in \mathbb{R}^d, \mathbf{u}_{t+1} \in \mathcal{C}_\rho, \mathbf{v}_{t+1} \in \mathcal{F}$. Let $\mathbf{w}_{t+1} = \mathbf{w}_t + \eta_t(\mathbf{u}_{t+1} - \mathbf{v}_{t+1})$. Then,*

$$
\begin{aligned}
\Delta_{t+1} - \Delta_t \leq \mathcal{L}(\mathbf{u}_{t+2}, \mathbf{v}_{t+2}, \mathbf{w}_{t+1}) - \mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) + \frac{2\eta_t}{\lambda}\left(\mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) - \mathcal{L}(\widehat{\mathbf{u}}_{t+1}, \widehat{\mathbf{v}}_{t+1}, \mathbf{w}_{t+1})\right) \\
- \frac{\eta_t \alpha^2}{2L_\lambda \zeta^2} \min\left\{\Delta^{(\mathrm{d})}_{t+1}, \frac{L_\lambda \zeta^2}{2}\right\}
\end{aligned}
$$

where $\alpha > 0$ is as defined in Section A.1.2. (Also Theorem 1 of Gidel et al. (2018) [15])

Proof. Let $\widehat{\mathbf{u}}_t, \widehat{\mathbf{v}}_t \in \operatorname{argmin}_{\mathbf{u} \in \mathcal{C}_\rho, \mathbf{v} \in \mathcal{F}} \mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{w}_t)$. We have that,

$$
\begin{aligned}
\Delta^{(\mathrm{d})}_{t+1} - \Delta^{(\mathrm{d})}_t &= \xi(\mathbf{w}_t) - \xi(\mathbf{w}_{t+1}) \\
&= \mathcal{L}(\widehat{\mathbf{u}}_t, \widehat{\mathbf{v}}_t, \mathbf{w}_t) - \mathcal{L}(\widehat{\mathbf{u}}_{t+1}, \widehat{\mathbf{v}}_{t+1}, \mathbf{w}_{t+1}) \\
&\leq \mathcal{L}(\widehat{\mathbf{u}}_{t+1}, \widehat{\mathbf{v}}_{t+1}, \mathbf{w}_t) - \mathcal{L}(\widehat{\mathbf{u}}_{t+1}, \widehat{\mathbf{v}}_{t+1}, \mathbf{w}_{t+1}) \\
&= \langle \mathbf{w}_t - \mathbf{w}_{t+1}, \widehat{\mathbf{u}}_{t+1} - \widehat{\mathbf{v}}_{t+1} \rangle \\
&= -\eta_t \langle \mathbf{u}_{t+1} - \mathbf{v}_{t+1}, \widehat{\mathbf{u}}_{t+1} - \widehat{\mathbf{v}}_{t+1} \rangle
\end{aligned}
$$

$$
\begin{aligned}
\Delta^{(\mathrm{p})}_{t+1} - \Delta^{(\mathrm{p})}_t &= \Delta^{(\mathrm{p})}(\mathbf{u}_{t+2}, \mathbf{v}_{t+2}, \mathbf{w}_{t+1}) - \Delta^{(\mathrm{p})}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_t) \\
&= \mathcal{L}(\mathbf{u}_{t+2}, \mathbf{v}_{t+2}, \mathbf{w}_{t+1}) - \mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_t) + \xi(\mathbf{w}_t) - \xi(\mathbf{w}_{t+1}) \\
&= \mathcal{L}(\mathbf{u}_{t+2}, \mathbf{v}_{t+2}, \mathbf{w}_{t+1}) - \mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) + \langle \mathbf{w}_{t+1} - \mathbf{w}_t, \mathbf{u}_{t+1} - \mathbf{v}_{t+1} \rangle + \xi(\mathbf{w}_t) - \xi(\mathbf{w}_{t+1}) \\
&= \mathcal{L}(\mathbf{u}_{t+2}, \mathbf{v}_{t+2}, \mathbf{w}_{t+1}) - \mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) + \eta_t \|\mathbf{u}_{t+1} - \mathbf{v}_{t+1}\|^2 + \xi(\mathbf{w}_t) - \xi(\mathbf{w}_{t+1})
\end{aligned}
$$

Putting both the bounds together, we get,

$$
\begin{aligned}
&\Delta_{t+1} - \Delta_t \\
&\leq \mathcal{L}(\mathbf{u}_{t+2}, \mathbf{v}_{t+2}, \mathbf{w}_{t+1}) - \mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) + \eta_t ||\mathbf{u}_{t+1} - \mathbf{v}_{t+1}||^2 - 2\eta_t \langle \mathbf{u}_{t+1} - \mathbf{v}_{t+1}, \widehat{\mathbf{u}}_{t+1} - \widehat{\mathbf{v}}_{t+1} \rangle \\
&= \mathcal{L}(\mathbf{u}_{t+2}, \mathbf{v}_{t+2}, \mathbf{w}_{t+1}) - \mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) + \eta_t ||(\mathbf{u}_{t+1} - \mathbf{v}_{t+1}) - (\widehat{\mathbf{u}}_{t+1} - \widehat{\mathbf{v}}_{t+1})||^2 - \eta_t ||\widehat{\mathbf{u}}_{t+1} - \widehat{\mathbf{v}}_{t+1}||^2 \\
&\leq \mathcal{L}(\mathbf{u}_{t+2}, \mathbf{v}_{t+2}, \mathbf{w}_{t+1}) - \mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) + \frac{2\eta_t}{\lambda}\left(\mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) - \mathcal{L}(\widehat{\mathbf{u}}_{t+1}, \widehat{\mathbf{v}}_{t+1}, \mathbf{w}_{t+1})\right) \\
&\quad - \eta_t ||\widehat{\mathbf{u}}_{t+1} - \widehat{\mathbf{v}}_{t+1}||^2 \qquad (14)
\end{aligned}
$$

The last inequality above follows from Lemma 10.

We have that $\widehat{\mathbf{u}}_{t+1} - \widehat{\mathbf{v}}_{t+1} = \nabla \xi(\mathbf{w}_{t+1})$, and by Theorem 1 of Gidel et al. [15] (also in Section A.1.2), we have that

$$
||\nabla \xi(\mathbf{w}_{t+1})||^2 \geq \frac{\alpha^2}{2L_\lambda \zeta^2} \min\left\{\Delta^{(\mathrm{d})}_{t+1}, \frac{L_\lambda \zeta^2}{2}\right\}.
$$

Putting it together we get

$$
\begin{aligned}
\Delta_{t+1} - \Delta_t \leq \mathcal{L}(\mathbf{u}_{t+2}, \mathbf{v}_{t+2}, \mathbf{w}_{t+1}) - \mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) + \frac{2\eta_t}{\lambda}\left(\mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) - \mathcal{L}(\widehat{\mathbf{u}}_{t+1}, \widehat{\mathbf{v}}_{t+1}, \mathbf{w}_{t+1})\right) \\
- \frac{\eta_t \alpha^2}{2L_\lambda \zeta^2} \min\left\{\Delta^{(\mathrm{d})}_{t+1}, \frac{L_\lambda \zeta^2}{2}\right\}
\end{aligned}
$$

□

We use Lemma 13 to prove Lemma 9. The proof of Lemma 13 closely follows the proof of Theorem 2 in [15] and is split into Lemmas 14, 15, 17. However, we make the iterates $C[h_t], \mathbf{v}_t$ over the set $\mathcal{C}$ and $\mathcal{F}$ explicit and derive results taking into account the approximate `LMO` for the set $\mathcal{C}$.

Lemma 13.

$$
\Delta_{t+1} - \Delta_t \leq -\frac{2}{t+2} \min(\Delta_{t+1}, \theta_1) + \frac{\theta_2}{(t+2)^2} + \frac{\theta_3}{t+2}(\rho + \rho')
$$

where $\theta_1 = \frac{L_\lambda \zeta^2}{2}$, $\theta_2 = 32 \frac{\beta_\lambda \zeta^2}{\chi^2 \lambda^2}\left(1 + \frac{2}{\chi \lambda}\right)$ and $\theta_3 = \frac{8}{\chi \lambda}\left(1 + \frac{2}{\chi \lambda}\right) \max_t \|\mathbf{a}_{t+1}\|$.

Proof. Lemma 12 and Lemma 11 leads to the following equation holding for $\gamma \in [0, 1]$,

$$\begin{aligned}
\Delta_{t+1} - \Delta_t &\leq \mathcal{L}(\mathbf{u}_{t+2}, \mathbf{v}_{t+2}, \mathbf{w}_{t+1}) - \mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) + \frac{2\eta_t}{\lambda}\left(\mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) - \mathcal{L}(\widehat{\mathbf{u}}_{t+1}, \widehat{\mathbf{v}}_{t+1}, \mathbf{w}_{t+1})\right) \\
&\quad - \frac{\eta_t \alpha^2}{2L_\lambda \zeta^2} \min\left\{\Delta^{(\mathrm{d})}_{t+1}, \frac{L_\lambda \zeta^2}{2}\right\} \\
&\leq \gamma_{t+2}\left(\mathcal{L}(\widehat{\mathbf{u}}_{t+1}, \widehat{\mathbf{v}}_{t+1}, \mathbf{w}_{t+1}) - \mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1})\right) + \gamma_{t+2}\|\mathbf{a}_{t+1}\|(\rho + \rho') + \frac{1}{2}\beta_\lambda \gamma_{t+2}^2 \zeta^2 \\
&\quad + \frac{2\eta_t}{\lambda}\left(\mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) - \mathcal{L}(\widehat{\mathbf{u}}_{t+1}, \widehat{\mathbf{v}}_{t+1}, \mathbf{w}_{t+1})\right) - \frac{\eta_t \alpha^2}{2L_\lambda \zeta^2} \min\left\{\Delta^{(\mathrm{d})}_{t+1}, \frac{L_\lambda \zeta^2}{2}\right\} \\
&= \left(\frac{2\eta_t}{\lambda} - \gamma_{t+2}\right)\left(\mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) - \mathcal{L}(\widehat{\mathbf{u}}_{t+1}, \widehat{\mathbf{v}}_{t+1}, \mathbf{w}_{t+1})\right) + \gamma_{t+2}\|\mathbf{a}_{t+1}\|(\rho + \rho') + \frac{1}{2}\beta_\lambda \gamma_{t+2}^2 \zeta^2 \\
&\quad - \frac{\eta_t \alpha^2}{2L_\lambda \zeta^2} \min\left\{\Delta^{(\mathrm{d})}_{t+1}, \frac{L_\lambda \zeta^2}{2}\right\}
\end{aligned}$$

Then for $\gamma_{t+2} = \frac{4\eta_t}{\lambda}$ we get,

$$\begin{aligned}
\Delta_{t+1} - \Delta_t &\leq -\left(\frac{2\eta_t}{\lambda}\right)\left(\mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) - \mathcal{L}(\widehat{\mathbf{u}}_{t+1}, \widehat{\mathbf{v}}_{t+1}, \mathbf{w}_{t+1})\right) + \frac{4\eta_t}{\lambda}\|\mathbf{a}_{t+1}\|(\rho + \rho') + \beta_\lambda \frac{8\eta_t^2}{\lambda^2}\zeta^2 \\
&\quad - \frac{\eta_t \alpha^2}{2L_\lambda \zeta^2} \min\left\{\Delta^{(\mathrm{d})}_{t+1}, \frac{L_\lambda \zeta^2}{2}\right\} \qquad (15)
\end{aligned}$$

We also have that,

$$\begin{aligned}
\mathcal{L}(\mathbf{u}_{t+2}, \mathbf{v}_{t+2}, \mathbf{w}_{t+1}) &\leq \mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) + \langle \mathbf{a}_{t+1}, \mathbf{u}_{t+2} - \mathbf{u}_{t+1}\rangle + \langle \mathbf{b}_{t+1}, \mathbf{v}_{t+2} - \mathbf{v}_{t+1}\rangle \\
&\quad + \frac{\beta_\lambda}{2}(\|\mathbf{u}_{t+2} - \mathbf{u}_{t+1}\|^2 + \|\mathbf{v}_{t+2} - \mathbf{v}_{t+1}\|^2) \\
&= \mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) + \gamma_{t+2}\langle \mathbf{a}_{t+1}, \widetilde{\mathbf{u}}_{t+2} - \mathbf{u}_{t+1}\rangle + \gamma_{t+2}\langle \mathbf{b}_{t+1}, \widetilde{\mathbf{v}}_{t+2} - \mathbf{v}_{t+1}\rangle \\
&\quad + \frac{\beta_\lambda}{2}\gamma_{t+2}^2(\|\widetilde{\mathbf{u}}_{t+2} - \mathbf{u}_{t+1}\|^2 + \|\widetilde{\mathbf{v}}_{t+2} - \mathbf{v}_{t+1}\|^2) \\
&\leq \mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) + \gamma_{t+2}\|\mathbf{a}_{t+1}\|(\rho + \rho') + \frac{\beta_\lambda}{2}\gamma_{t+2}^2(\zeta^2) \qquad (16)
\end{aligned}$$

Rearrranging terms we get

$$-\mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) \leq -\mathcal{L}(\mathbf{u}_{t+2}, \mathbf{v}_{t+2}, \mathbf{w}_{t+1}) + \gamma_{t+2}\|\mathbf{a}_{t+1}\|(\rho + \rho') + \frac{\beta_\lambda}{2}\gamma_{t+2}^2 \zeta^2 \qquad (17)$$

Substituting Eqn. (17) in Eqn. (15), we get

$$\begin{aligned}
\Delta_{t+1} - \Delta_t &\leq -\left(\frac{2\eta_t}{\lambda}\right)\left(\mathcal{L}(\mathbf{u}_{t+2}, \mathbf{v}_{t+2}, \mathbf{w}_{t+1}) - \mathcal{L}(\widehat{\mathbf{u}}_{t+1}, \widehat{\mathbf{v}}_{t+1}, \mathbf{w}_{t+1}) - \frac{4\eta_t}{\lambda}\|\mathbf{a}_{t+1}\|(\rho + \rho') - \frac{8\eta_t^2 \beta_\lambda \zeta^2}{\lambda^2}\right) \\
&\quad + \frac{4\eta_t}{\lambda}\|\mathbf{a}_{t+1}\|(\rho + \rho') + \beta_\lambda \frac{8\eta_t^2}{\lambda^2}\zeta^2 - \frac{\eta_t \alpha^2}{2L_\lambda \zeta^2} \min\left\{\Delta^{(\mathrm{d})}_{t+1}, \frac{L_\lambda \zeta^2}{2}\right\} \\
&= -\left(\frac{2\eta_t}{\lambda}\right)\Delta^{\mathrm{p}}_{t+1} - \frac{\eta_t \alpha^2}{2L_\lambda \zeta^2} \min\left\{\Delta^{(\mathrm{d})}_{t+1}, \frac{L_\lambda \zeta^2}{2}\right\} + \left(\frac{4\eta_t}{\lambda}\|\mathbf{a}_{t+1}\|(\rho + \rho') + \beta_\lambda \frac{8\eta_t^2}{\lambda^2}\zeta^2\right)\left(1 + \frac{2\eta_t}{\lambda}\right) \\
&\leq -\chi\eta_t \Delta^{\mathrm{p}}_{t+1} - \chi\eta_t \min\left\{\Delta^{(\mathrm{d})}_{t+1}, \frac{L_\lambda \zeta^2}{2}\right\} + \left(\frac{4\eta_t}{\lambda}\|\mathbf{a}_{t+1}\|(\rho + \rho') + \beta_\lambda \frac{8\eta_t^2}{\lambda^2}\zeta^2\right)\left(1 + \frac{2\eta_t}{\lambda}\right) \\
&\leq -\chi\eta_t \min\left\{\Delta_{t+1}, \frac{L_\lambda \zeta^2}{2}\right\} + \left(\frac{4\eta_t}{\lambda}\|\mathbf{a}_{t+1}\|(\rho + \rho') + \beta_\lambda \frac{8\eta_t^2}{\lambda^2}\zeta^2\right)\left(1 + \frac{2\eta_t}{\lambda}\right)
\end{aligned}$$

where $\chi = \min\left\{\frac{2}{\lambda}, \frac{\alpha^2}{2\beta_\lambda \zeta^2}\right\}$. Letting $\eta_t = \frac{2}{\chi(t+2)}$, we get

$$\Delta_{t+1} - \Delta_t \leq -\frac{2}{t+2}\min\left\{\Delta_{t+1}, \frac{L_\lambda \zeta^2}{2}\right\} + \left(\frac{8}{\chi\lambda(t+2)}\|\mathbf{a}_{t+1}\|(\rho + \rho') + 32\frac{\beta_\lambda \zeta^2}{\chi^2\lambda^2(t+2)^2}\right)\left(1 + \frac{4}{\chi\lambda(t+2)}\right)$$

We thus have that,

$$\Delta_{t+1} - \Delta_t \leq -\frac{2}{t+2}\min(\Delta_{t+1}, \theta_1) + \frac{\theta_2}{(t+2)^2} + \frac{\theta_3}{t+2}(\rho + \rho') \qquad (18)$$

where $\theta_1 = \frac{L_\lambda \zeta^2}{2}$, $\theta_2 = 32\frac{\beta_\lambda \zeta^2}{\chi^2\lambda^2}\left(1 + \frac{2}{\chi\lambda}\right)$ and $\theta_3 = \frac{8}{\chi\lambda}\left(1 + \frac{2}{\chi\lambda}\right)\max_t \|\mathbf{a}_{t+1}\|$. □

From Lemma 8, we have that $\|\mathbf{w}_t\|$ is bounded by a constant if the duality gap Δ_t is bounded, and hence $\|\mathbf{a}_{t+1}\| = \|\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1})\| = \|\nabla\psi(\mathbf{u}_{t+1}) + \mathbf{w}_{t+1} + \lambda(\mathbf{u}_{t+1} - \mathbf{v}_{t+1})\|$ can also be bounded by a constant. We will need $2\theta_1 > \theta_3(\rho + \rho')$ for there to be a decrease in Δ_t, this can be simply achieved by setting λ to be a large enough value. Because if λ is large, $\chi \approx \frac{c}{\lambda}$, and hence θ_3 becomes a constant when increasing λ further, but θ_1 keeps increasing linearly with λ.

Lemma 14. *Let Δ_t be a sequence satisfying Eqn.* (18)*. Let $2\theta_1 > \theta_3(\rho + \rho')$. Let there exist a $t_0 > \frac{\theta_2}{2\theta_1 - \theta_3(\rho+\rho')} - 2$ such that $\Delta_{t_0} \leq \theta_1$, then*

$$\Delta_t \leq \min\left\{\frac{4\theta_1(t_0+2)}{t+2} + \frac{\theta_3(\rho+\rho')}{2},\ \theta_1\right\} \quad \forall t \geq t_0\ . \tag{19}$$

Proof. For $t = t_0$ the bound on Δ_t simplifies to θ_1 and hence is true. This will form our base case for proof by induction. We make the induction assumption that for a $t \geq t_0,\ \Delta_t \leq \min\left\{\frac{4\theta_1(t_0+2)}{t+2} + \frac{\theta_3(\rho+\rho')}{2},\ \theta_1\right\}$.

If $\Delta_{t+1} > \theta_1$, then

$$\begin{aligned}
\theta_1 < \Delta_{t+1} &\leq \Delta_t - \frac{2\theta_1}{t+2} + \frac{\theta_2}{(t+2)^2} + \frac{\theta_3}{t+2}(\rho+\rho') \\
\Delta_{t+1} &\leq \theta_1 - \frac{2\theta_1}{t+2} + \frac{\theta_2}{(t+2)^2} + \frac{\theta_3}{t+2}(\rho+\rho') \\
2\theta_1 - \theta_3(\rho+\rho') &< \frac{\theta_2}{t+2} \\
t &< \frac{\theta_2}{2\theta_1 - \theta_3(\rho+\rho')} - 2
\end{aligned}$$

which contradicts $t \geq t_0 > \frac{\theta_2}{2\theta_1 - \theta_3(\rho+\rho')} - 2$. Hence $\Delta_{t+1} < \theta_1$. Thus, from Eqn. (18), we have

$$\begin{aligned}
\Delta_{t+1} &\leq \Delta_t - \frac{2}{2+t}\Delta_{t+1} + \frac{\theta_2}{(t+2)^2} + \frac{\theta_3}{t+2}(\rho+\rho') \\
\Delta_{t+1}\frac{t+4}{t+2} &\leq \Delta_t + \frac{\theta_2}{(t+2)^2} + \frac{\theta_3}{t+2}(\rho+\rho') \\
\Delta_{t+1}\frac{t+4}{t+2} &\leq \frac{4\theta_1(t_0+2)}{t+2} + \frac{\theta_3(\rho+\rho')}{2} + \frac{\theta_2}{(t+2)^2} + \frac{\theta_3}{t+2}(\rho+\rho') \\
\Delta_{t+1} &\leq \frac{4\theta_1(t_0+2)}{t+4} + \frac{\theta_3(\rho+\rho')}{2}(\frac{t+2}{t+4} + \frac{2}{t+4}) + \frac{\theta_2}{(t+2)(t+4)} \\
\Delta_{t+1} &\leq \frac{4\theta_1(t_0+2)}{t+4} + \frac{\theta_3(\rho+\rho')}{2} + \frac{2\theta_1(t_0+2)}{(t+2)(t+4)} \\
\Delta_{t+1} &\leq 4\theta_1(t_0+2)\left(\frac{1}{t+4} + \frac{1}{2(t+2)(t+4)}\right) + \frac{\theta_3(\rho+\rho')}{2} \\
\Delta_{t+1} &\leq 4\theta_1(t_0+2)\left(\frac{1}{t+4} + \frac{1}{(t+3)(t+4)}\right) + \frac{\theta_3(\rho+\rho')}{2} \\
\Delta_{t+1} &\leq 4\theta_1(t_0+2)\left(\frac{1}{t+3}\right) + \frac{\theta_3(\rho+\rho')}{2}
\end{aligned}$$

And hence, we have

$$\Delta_{t+1} \leq \min\left\{\frac{4\theta_1(t_0+2)}{t+3} + \frac{\theta_3(\rho+\rho')}{2}, \theta_1\right\}$$

The Lemma thus holds by induction. □

Now we have to show that in a constant number of iterations t_0 we can reach a point such that $\Delta_{t_0} \leq \theta$.

Lemma 15. *Let Δ_t be a sequence satisfying Eqn.* (18)*. Let $2\theta_1 > \theta_3(\rho + \rho')$. Then there exists a constant $t_0 > \frac{\theta_2}{2\theta_1 - \theta_3(\rho+\rho')} - 2$ such that $\Delta_{t_0} \leq \theta_1$.*

Proof. Clearly, there must exist a t_0 such that $\Delta_{t_0} \leq \theta_1$, because $\frac{1}{t}$ is a divergent series, and $\frac{1}{t^2}$ is a convergent series and Δ_t is bounded below by 0.

The same argument can be used for saying that Δ_t drops below θ_1 infinitely often, and hence there exists $t_0 > \frac{\theta_2}{2\theta_1 - \theta_3(\rho+\rho')} - 2$ such that $\Delta_{t_0} \leq \theta_1$.

Let t_0 be the first instant $t > \frac{\theta_2}{2\theta_1 - \theta_3(\rho+\rho')} - 2$ such that $\Delta_t \le \theta_1$, clearly this t_0 can be upper bounded by a constant that depends only on the two numbers $2\theta_1 - \theta_3(\rho+\rho')$ and θ_2. $\square$

Putting together Lemmas 13, 14 and 15, we get the following corollary.

Corollary 16. *Let $2\theta_1 > \theta_3(\rho+\rho')$. There exists a constant $t_0 > 0$ such that*

$$\Delta_t \le \frac{4\theta_1(t_0+2)}{t+2} + \frac{\theta_3(\rho+\rho')}{2} \quad \forall t \ge t_0 \ .$$

where $\theta_1 = \frac{L_\lambda \zeta^2}{2}$ and $\theta_3 = \frac{8}{\chi\lambda}\left(1 + \frac{2}{\chi\lambda}\right)\max_t \|\mathbf{a}_{t+1}\|$.

Lemma 17. *Let $2\theta_1 > \theta_3(\rho+\rho')$. Let $t_0 \in \mathbb{N}$ be as in Lemma 15. Let $\mathbf{u}_t, \mathbf{v}_t, \mathbf{w}_t$ be as in Algorithm 2. Then for all $T > 2t_0$ and $T > 10$, there exists a $t \in [T/2, T]$ such that*

$$||\mathbf{u}_t - \mathbf{v}_t||^2 \le \frac{c}{T} + c(\rho+\rho')$$

for some constant $c > 0$.

Proof. Rewriting Eqn. (14) here, we have

$$\begin{aligned}
&\Delta_{t+1} - \Delta_t \\
&\le \mathcal{L}(\mathbf{u}_{t+2}, \mathbf{v}_{t+2}, \mathbf{w}_{t+1}) - \mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) + \frac{2\eta_t}{\lambda}\left(\mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) - \mathcal{L}(\widehat{\mathbf{u}}_{t+1}, \widehat{\mathbf{v}}_{t+1}, \mathbf{w}_{t+1})\right) \\
&\quad - \eta_t||\widehat{\mathbf{u}}_{t+1} - \widehat{\mathbf{v}}_{t+1}||^2
\end{aligned}$$

With the above equation as the starting point and proceeding as we do in Lemma 13, we get the below inequality that is similar to Eqn. (15)

$$\begin{aligned}
\Delta_{t+1} - \Delta_t \le& -\left(\frac{2\eta_t}{\lambda}\right)\left(\mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) - \mathcal{L}(\widehat{\mathbf{u}}_{t+1}, \widehat{\mathbf{v}}_{t+1}, \mathbf{w}_{t+1})\right) + \frac{4\eta_t}{\lambda}\|\mathbf{a}_{t+1}\|(\rho+\rho') + \beta_\lambda \frac{8\eta_t^2}{\lambda^2}\zeta^2 \\
&- \eta_t||\widehat{\mathbf{u}}_{t+1} - \widehat{\mathbf{v}}_{t+1}||^2
\end{aligned}$$

Let $h_{t+1} = (\mathcal{L}(C[h_{t+1}], \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) - \mathcal{L}(\widehat{\mathbf{u}}_{t+1}, \widehat{\mathbf{v}}_{t+1}, \mathbf{w}_{t+1})) \ge 0$. We then have

$$\frac{2}{\lambda}\left(h_{t+1} - 2\|\mathbf{a}_{t+1}\|(\rho+\rho')\right) + ||\widehat{\mathbf{u}}_{t+1} - \widehat{\mathbf{v}}_{t+1}||^2 \le \frac{\Delta_t - \Delta_{t+1}}{\eta_t} + \eta_t \frac{8\beta_\lambda\zeta^2}{\lambda^2} \tag{20}$$

Let the dual step size $\eta_t = \frac{2}{\chi(t+2)}$. Let $\{w_t\}_{T/2}^{T}$ be a sequence of positive weights. We set $w_t = t - T/2$. Let $\tau_t = \frac{w_t}{\sum_{t=\frac{T}{2}}^{T} w_t} = \frac{2t-T}{(T/2)(T/2+1)}$ be the associated normalized weights. The convex combination of Eqn. (20)

with weights τ_t gives us

$$\begin{aligned}
&\sum_{t=T/2}^{T} \tau_t \left(\frac{2}{\lambda} \left(h_{t+1} - 2\|\mathbf{a}_{t+1}\|(\rho + \rho') \right) + ||\widehat{\mathbf{u}}_{t+1} - \widehat{\mathbf{v}}_{t+1}||^2 \right) \\
&\leq \sum_{t=T/2}^{T} \tau_t \frac{\Delta_t - \Delta_{t+1}}{\eta_t} + \sum_{t=T/2}^{T} \tau_t \eta_t \frac{8\beta_\lambda \zeta^2}{\lambda^2} \\
&= \frac{\tau_{T/2}}{\eta_{T/2}} \Delta_{T/2} - \frac{\tau_T}{\eta_T} \Delta_T + \sum_{t=T/2+1}^{T} \Delta_t \left(\frac{\tau_{t+1}}{\eta_{t+1}} - \frac{\tau_t}{\eta_t} \right) + \sum_{t=T/2}^{T} \tau_t \eta_t \frac{8\beta_\lambda \zeta^2}{\lambda^2} \\
&\leq \sum_{t=T/2+1}^{T} \left(\frac{4\theta_1(t_0+2)}{t+2} + \frac{\theta_3(\rho+\rho')}{2} \right) \left(\frac{\tau_{t+1}}{\eta_{t+1}} - \frac{\tau_t}{\eta_t} \right) + \sum_{t=T/2}^{T} \tau_t \eta_t \frac{8\beta_\lambda \zeta^2}{\lambda^2} \\
&= \sum_{t=T/2+1}^{T} \left(\frac{4\theta_1(t_0+2)}{t+2} \right) \left(\frac{8t - 2T + 12}{\chi T(T+2)} \right) + \left(\frac{\theta_3(\rho+\rho')}{2} \right) \left(\frac{\tau_T}{\eta_T} \right) + \sum_{t=T/2}^{T} \tau_t \eta_t \frac{8\beta_\lambda \zeta^2}{\lambda^2} \\
&\leq \sum_{t=T/2+1}^{T} \left(\frac{4\theta_1(t_0+2)}{T/2+2} \right) \left(\frac{8T - 2T + 12}{\chi T(T+2)} \right) + \left(\frac{\theta_3(\rho+\rho')}{2} \right) \left(\frac{4(T+2)}{T2\chi} \right) + \sum_{t=T/2}^{T} \tau_T \eta_{T/2} \frac{8\beta_\lambda \zeta^2}{\lambda^2} \\
&\leq \sum_{t=T/2+1}^{T} \left(\frac{8\theta_1(t_0+2)}{T} \right) \left(\frac{12}{\chi T} \right) + \left(\frac{\theta_3(\rho+\rho')}{2} \right) \left(\frac{8}{2\chi} \right) + \sum_{t=T/2}^{T} \frac{4}{T} \frac{4}{\chi T} \frac{8\beta_\lambda \zeta^2}{\lambda^2} \\
&\leq \frac{48\theta_1(t_0+2)}{\chi T} + \frac{2\theta_3(\rho+\rho')}{\chi} + \frac{64\beta_\lambda \zeta^2}{\chi \lambda^2 T}
\end{aligned}$$

Thus, there must exist a $t \in [T/2, T]$ such that

$$\frac{2h_{t+1}}{\lambda} + ||\widehat{\mathbf{u}}_{t+1} - \widehat{\mathbf{v}}_{t+1}||^2 \leq \frac{48\lambda^2\theta_1(t_0+2) + 64\beta_\lambda\zeta^2}{\chi\lambda^2 T} + \left(\frac{2\theta_3}{\chi} + \frac{4}{\lambda} \max_t \|\mathbf{a}_t\| \right) (\rho + \rho'). \tag{21}$$

Now from Lemma 10 we have

$$\|(\mathbf{u}_{t+2} - \mathbf{v}_{t+2}) - (\widehat{\mathbf{u}}_{t+1} - \widehat{\mathbf{v}}_{t+1})\|^2 \leq \frac{2}{\lambda} \left(\mathcal{L}(\mathbf{u}_{t+2}, \mathbf{v}_{t+2}, \mathbf{w}_{t+1}) - \mathcal{L}(\widehat{\mathbf{u}}_{t+1}, \widehat{\mathbf{v}}_{t+1}, \mathbf{w}_{t+1}) \right) = \frac{2}{\lambda} \Delta_t^{(\mathrm{p})} \tag{22}$$

From Eqn. (16), we have

$$\begin{aligned}
\Delta_{t+1}^{(\mathrm{p})} - h_{t+1} &= \mathcal{L}(\mathbf{u}_{t+2}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) - \mathcal{L}(\mathbf{u}_{t+1}, \mathbf{v}_{t+1}, \mathbf{w}_{t+1}) \\
&\leq \frac{4\eta_t}{\lambda} \|\mathbf{a}_{t+1}\|(\rho + \rho') + \frac{32\beta_\lambda\zeta^2}{\lambda^2\chi^2(t+2)^2}
\end{aligned} \tag{23}$$

Putting Eqns. (21), (22) and (23), we get Thus, there must exist a $t \in [T/2, T]$ such that

$$||\mathbf{u}_t - \mathbf{v}_t||^2 \leq \frac{48\lambda^2\theta_1(t_0+2) + 64\beta_\lambda\zeta^2 + 16\lambda \max_t \|\mathbf{a}_t\|(\rho+\rho')}{\chi\lambda^2 T} + \left(\frac{2\theta_3}{\chi} + \frac{4}{\lambda} \max_t \|\mathbf{a}_t\| \right) (\rho + \rho') + O\left(\frac{1}{T^2} \right). \tag{24}$$

□

B Further Related Work

B.1 The COCO Approach

We elaborate on the COCO approach of Narasimhan (2018) [29], which is statistically consistent, but is shown to achieve only a $O(1/\epsilon^3)$ convergence rate. Like us, this approach also reformulates (OP1) as an optimization over $\mathcal{C}$ but retains explicit constraints $\phi_k(C) \leq 0$, $\forall k$:

$$\begin{aligned}
&\min_{C \in \mathcal{C}} \psi(C) \\
&\text{s.t. } \phi_k(C) \leq 0, \ \forall k \in [K].
\end{aligned}$$

The idea is to then formulate the Lagrangian for the constrained problem with Lagrange multipliers $\lambda \in \Lambda \subset \mathbb{R}_+^K$:

$$\mathcal{L}(C, \lambda) \,=\, \psi(C) \,+\, \sum_{k=1}^{K} \lambda_k \phi_k(C),$$

and to maximize the Lagrangian over the multipliers using gradient ascent:

$$C^{(t+1)} \;\in\; \operatorname*{argmin}_{C \in \mathcal{C}} \mathcal{L}(C, \lambda^{(t)}) \tag{25}$$

$$\lambda^{t+1} \;=\; \Pi_\Lambda\left(\lambda^{(t)} \,-\, \nabla_\lambda \mathcal{L}(C^{(t+1)} \lambda^{(t)})\right), \tag{26}$$

where Π_Λ is the projection onto the set Λ.

Note, however, that each gradient update on λ requires a minimization of the Lagrangian over $\mathcal{C}$ in (25), and this is performed with a full run of the classical Frank-Wolfe method [18] using calls to a plug-in routine to solve the LMO needed in each iteration. The final algorithm has two levels of nesting, where the inner level solves the minimization in (25) with $O(1/\epsilon)$ calls to the plug-in routine, and the outer level performs $O(1/\epsilon^2)$ gradient ascent steps, resulting in a total $O(1/\epsilon^3)$ calls to the plug-in routine to reach an ϵ-optimal, ϵ-feasible solution.

B.2 The 3-player Approach

As noted in Section 1, Narasimhan et al. (2019) [30] provide an idealized algorithm that enjoys the same convergence rate as our approach to the optimal feasible solution, but do not provide a full-fledged consistency analysis for this method. We elaborate on this method below.

Under the assumption that ψ and ϕ_k's are monotonically non-decreasing in their arguments, Narasimhan et al. reformulate (OP1) by introducing slack variables $\xi \in [0, 1]^d$ and arrive at the following equivalent problem:

$$\begin{aligned} &\min_{h \in \mathcal{H},\, \xi \in [0,1]^d} \psi(\xi) \\ \text{s.t. }\; &\phi_k(\xi) \leq 0,\, \forall k \in [K] \\ &\xi \geq C[h] \end{aligned}$$

They then formulate the Lagrangian for this problem with multipliers $\lambda \in \Lambda \subset \mathbb{R}_+^{K+d}$:

$$\mathcal{L}(h, \xi, \lambda) \,=\, \psi(\xi) \,+\, \sum_{k=1}^{K} \lambda_k \phi_k(\xi) \,+\, \sum_{i=1}^{d} \lambda_{K+i} \left(C_i[h] \,-\, \xi_i\right),$$

and perform the following sequence of updates at each step t:

$$h^{(t+1)} \;\in\; \operatorname*{argmin}_{h \in \mathcal{H}} \mathcal{L}(h, \xi^{(t)}, \lambda^{(t)}) \tag{27}$$

$$\xi^{(t+1)} \;\in\; \operatorname*{argmin}_{\xi \in [0,1]^d} \mathcal{L}(h^{(t)}, \xi, \lambda^{(t)}) \tag{28}$$

$$\lambda^{t+1} \;=\; \Pi_\Lambda\left(\lambda^{(t)} \,-\, \nabla_\lambda \mathcal{L}(h^{(t+1)}, \xi^{(t+1)} \lambda^{(t)})\right), \tag{29}$$

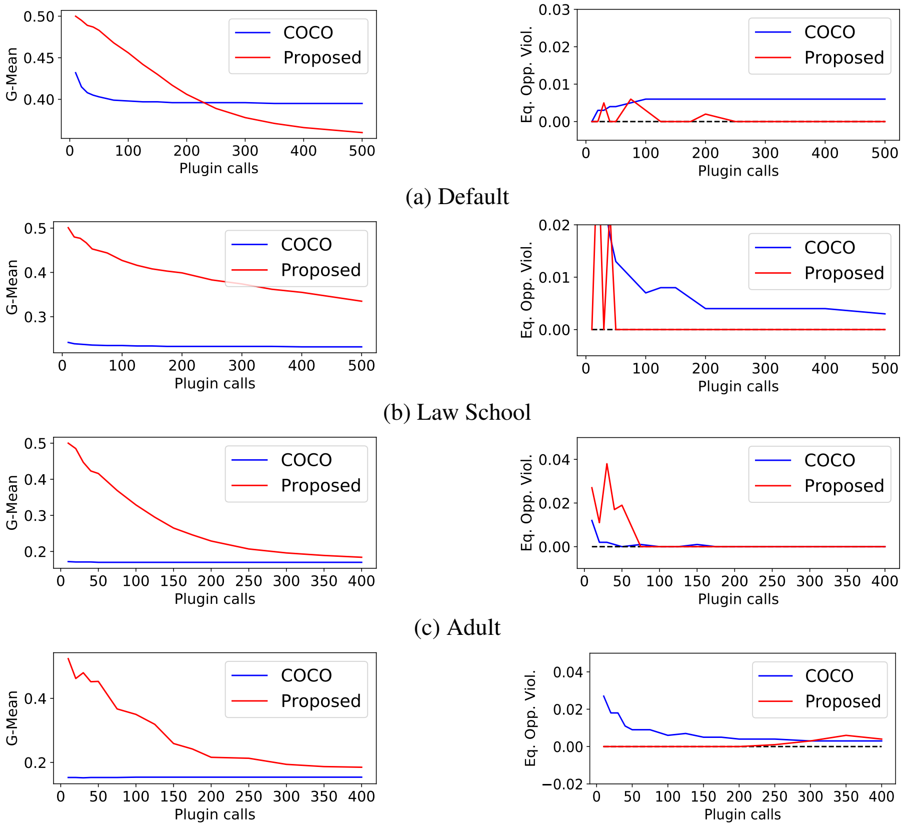
where Π_Λ is the projection onto the set Λ.

The authors then show that when ψ and ϕ_k's are convex, these updates converge to an ϵ-optimal, ϵ-feasible classifier after $O(1/\epsilon^2)$ steps. However, this result relies on access to an oracle for performing the optimization in (27) over the space of classifiers $\mathcal{H}$ near-optimally. The authors further acknowledge that such an oracle may not exist for general settings, and prescribe a more 'practical' algorithm that replaces (27) with a gradient update on a relaxed Lagrangian objective, but does not enjoy the same convergence guarantees. We compare against this surrogate-based approach (referred to as the 3-player method) in the experiments in Section 5. In the open-source implementation the authors provide [10], they further replace (28) with a gradient update on ξ.

The algorithm we propose in this paper also uses two minimization subroutines, namely an LMO over $\mathcal{C}$ and an LMO over $\mathcal{F}$, but both of these can be implemented efficiently. The LMO over $\mathcal{C}$ is implemented in our approach using a plug-in classifier, and the LMO over $\mathcal{F}$ reduces to a simple convex program and can often be implemented very efficiently with a specialized solver. Unlike the the 3-player approach, we do not maintain an explicit Lagrange multiplier for each constraint, and access the constraint set only through an LMO. In Section 4, we then provide a complete consistency analysis, showing optimality and feasibility bounds in terms of the quality of class-probability estimates used to implement the plug-in classifier. Our results, however, require the objective function ψ to be smooth, whereas Narasimhan et al. do not require this.

Table 4: Datasets used in our experiments.

Dataset	Instances	Features	Protected Attribute	Prot. group frac.
COMPAS	6172	32	Gender	0.19
Communities & Crime	1994	132	Race	0.49
Law School	20798	16	Race	0.06
Adult	48842	123	Gender	0.10
Default	30000	23	Gender	0.40



(a) Default

(b) Law School

(c) Adult

(d) Crime

Figure 4: Training objective (left) and constraint violation (right) as function of no. of plug-in calls for the task of minimizing G-mean subject to an equal opportunity constraint.

C Additional Experimental Details

Table 4 lists the datasets used in our experiments. Figure 4 shows convergence of the proposed method and the prior COCO method as function of the number of calls to the plug-in method. Figure 5 demonstrates robustness of our approach to hyper-parameter choices.

Hyper-parameters. To implement the plug-in routine, we use a pre-trained linear logistic regression model to estimate $\widehat{\boldsymbol{\eta}}$, with the protected attribute included as one of the features. For the problems we consider in the experiments, the LMO over the feasible set $\mathcal{F}$ in the proposed SBFW method is a linear program (LP), which we solve using a standard LP solver. For the proposed SBFW, we set η_t as a decreasing step function with values $\{0.5, 0.1, 0.001\}$, set $\lambda = 10$ and $\gamma_t = \frac{2}{t+2}$. For COCO, we tuned the learning rates from the range $\{0.01, 0.1, 0.5, 1, 10, 20\}$. For the 3-player approach, we tuned the learning rates for the model and constraint from $\{0.01, 0.1, 0.5, 1\}$. We note that SBFW requires almost no tuning compared to COCO and 3-player for the experiments. In each case, we pick the best hyper-parameter using a heuristic provided by Cotter et al. (2019) [10] to find the best trade-off between the training objective and constraint violations. We ran the 3-player

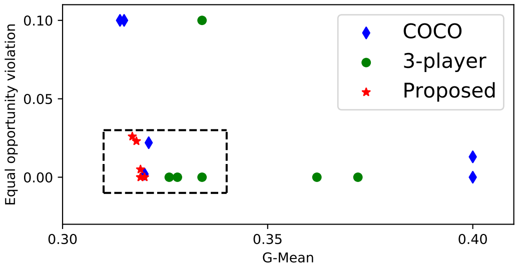


Figure 5: Robustness to hyper-parameters: Scatter plot of train G-mean and equal opportunity violation (with negative values clipped to zero) for six step sizes. While all six choices achieved close-to-best objective and near-zero violations for the proposed algorithm, only two choices led to similar metrics for COCO, and three choices led to similar metrics for the 3-player method.

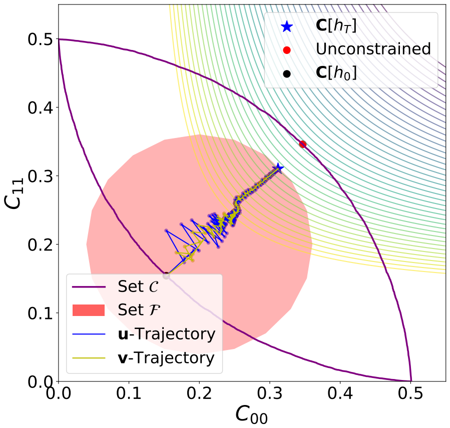


Figure 6: An illustration of Algorithm 1 for a synthetic 2-class problem with 20 constraints. We consider a data distribution with equal prior probabilities and with class conditionals $X|Y = 0$ and $X|Y = 1$ distributed as a standard normal with means 1 and 0 respectively. The goal is to minimize H-mean subject to a 20-sided polygonal constraint.

method for 2000 iterations and ran COCO and SBFW with 2000 calls to the plug-in routine. In the experiments in Figure 4, we separately tune the hyper-parameters for each method. For all experiments, we measure the constraint violation by the positive part of $\phi(h) - \epsilon$, that is using $\max\{0, \phi(h) - \epsilon\}$.

Larger number of constraints. In our final experiment, we demonstrate the effectiveness of the proposed approach in handling a larger number constraints than considered in Section 5. For this, we consider a synthetic 2-class problem, with equal prior probabilities and with class conditionals $X|Y = 0$ and $X|Y = 1$ distributed as a standard normal with means $+1$ and 0 respectively. The goal is to minimize H-mean subject to the diagonal entries of the confusion matrix that the confusion matrix lies within a polygon centered at $(0.2, 0.2)$. Figure 6 shows that contours of the objective function and the polygonal constraint region highlighted in red. The polygon is represented by 20 linear constraints. We find that the proposed method converges to a near-optimal, near-feasible solution within 40 calls to the plug-in routine.