Learning with Complex Loss Functions and Constraints

Harikrishna Narasimhan
Institute for Applied Computational Science, SEAS, Harvard University, Cambridge, MA 02138, USA

Abstract

We develop a general approach for solving constrained classification problems, where the loss and constraints are defined in terms of a general function of the confusion matrix. We are able to handle complex, non-linear loss functions such as the F-measure, G-mean or H-mean, and constraints ranging from budget limits, to constraints for fairness, to bounds on complex evaluation metrics. Our approach builds on the framework of Narasimhan et al. (2015) for unconstrained classification with complex losses, and reduces the constrained learning problem to a sequence of cost-sensitive learning tasks. We provide algorithms for two broad families of problems, involving convex and fractional-convex losses, subject to convex constraints. Our algorithms are statistically consistent, generalize an existing approach for fair classification, and readily apply to multiclass problems. Experiments on a variety of tasks demonstrate the efficacy of our methods.

1 INTRODUCTION

In numerous prediction tasks, one is required to learn a classifier that optimizes a loss function, subject to a set of constraints. The following are some constrained classification problems that arise in the real-world:

- **Precision/budget constraints:** In applications such as information retrieval, one often wishes to maximize the recall of a classifier subject to its precision being above a certain limit. In applications where there is a monetary cost associated with a positive prediction, one might want to optimize classification performance subject to a budget on the fraction of positive predictions.
- **Fairness constraints:** In applications involving social data (e.g. college admissions), it is important that the learned classifier is fair across different social groups. This results in a set of constraints on the classifier, such as, the proportion of positive predictions being similar for all subgroups, or the proportion of true/false positives being similar for all subgroups [20, 12].
- **Quantification constraints:** The problem of quantification aims at estimating the prevalance/frequency of a class in a population [8]. This arises, for e.g. in sentiment analysis, where in addition to predicting the sentiment for individual articles, one often wishes to estimate the overall prevalence of a sentiment in the population. This results in a classification problem with two competing objectives: (a) the classification performance of the model and (b) the quantification error of the model, usually measured by the KL-divergence (or ℓ_1 error) between the estimated class distribution and the true class distribution [7, 10]. One way to formulate this problem is to optimize the classification loss subject to the quantification error being within a threshold.
- **Churn constraints:** In applications where a classifier needs to be improved over time, it is desirable that the new classifier does not differ from an existing classifier on more than a certain proportion of examples, i.e. the churn rate of the classifier is within a desired limit [11].

In the above examples, the performance measure that one seeks to optimize need not be a simple error metric that can be expressed as a sum or expectation of point-wise losses. For example in many retrieval tasks, it is desirable to optimize the F_1-measure [23, 19]; in classification tasks with class imbalance, a classifier is often evaluated using metrics such as the G-mean or the H-mean measure that emphasize equitable performance on all classes [30, 32, 22]. These evaluation measures are complex, non-linear functions of the confusion matrix of the classifier. Similarly, the constraints imposed on the classifier may also have a complex structure. E.g. in the quantification example, the KL-divergence is a non-linear function of the confusion matrix, and does not decompose into a sum of point-wise terms.

Our focus is on a general classification problem with a loss $L(h)$ that is possibly a non-linear function of the confusion matrix of classifier h, and with constraints $g_k(h) \leq \epsilon_k,\ \forall k = 1, \ldots K$, where each g_k is a function of the confusion matrix of h.

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

Previous approaches for constrained classification handle a simpler version of the above problem where the loss and constraint functions are linear in the confusion matrix [11, 1]. Much of these works have focused on learning classifiers under constraints for fairness [12, 20]. A common approach here is to either introduce a regularization term that penalizes violations in constraints, or to formulate an unconstrained weighted version of the problem, and replace the non-continuous portions of the objective and constraints with continuous surrogate functions [15, 3, 33, 4]. These approaches are tailored to specific fairness metrics, and do not generalize to problems where the loss or constraint functions are complex and non-decomposable.

In a recent work that departs from the surrogate/regularization route, Agarwal et al. [1] point out that for two popular fairness metrics, the fairness-constrained classification problem can be reduced to cost-sensitive classification tasks. However, their approach does not apply to general, non-linear losses and constraints.

In this paper, we develop a generic approach for constrained classification that enjoys several advantages. Firstly, our approach applies to a broad family of complex loss functions and constraints used widely in practice. Secondly, our approach does not use convex relaxations, and instead proceeds by formulating a sequence of cost-sensitive learning problems. Thirdly, the algorithms developed come with provable consistency guarantees. Finally, our approach readily applies to multiclass classification settings, with running time polynomial in the number of classes. This is in contrast to common learning algorithms for unconstrained complex losses (e.g. SVMPerf [14]), which are intractable for large multiclass problems.

Our approach builds on the framework of Narasimhan et al. (2015) for unconstrained problems with complex loss functions [27]. The key idea is to formulate the problem of constrained classification as an optimization problem over the space of confusion matrices, and to employ optimization solvers that operate through a sequence of linear minimization steps. The individual steps reduce to cost-sensitive classification problems, with the final classifier being a randomized classifier that combines multiple plug-in classifiers (e.g. cost-weighted logistic regression).

We provide algorithms for two families of constrained classification problems that cover all the examples seen above: (a) loss functions that are convex functions of the confusion matrix (e.g. G-mean or H-mean used in class-imbalanced problems), under constraint functions that are convex in the confusion matrix; and (b) loss functions that are fractional-convex functions of the confusion matrix (e.g. F_1 measure in text retrieval), under convex constraint functions.

The proposed algorithms are *statistically consistent*, i.e. converge in the large sample limit to the optimal loss and to zero constraint violation. As a byproduct of our results, we provide consistent algorithms for classification under constraints for fairness. To the best of our knowledge, this is the first such result in the fair classification literature.

Related Work. The last few years has seen much work on algorithms and theory for complex loss functions. These include the SVMPerf method [14], plug-in classifiers [28, 21], and online optimization methods for non-decomposable losses [18, 17, 26, 29, 16]. None of these methods however apply to settings with constraints. Some recent results in the fairness literature characterize the form of the optimal classifier for different fairness metrics, but again consider only linear losses [12, 6, 25]. The most general work that we are aware of on constrained classification is that of Goh et al. (2016) [11], who provide a surrogate-style approach for optimizing linear losses with linear constraints. Their approach requires solving a non-convex optimization problem, and does not extend to complex losses. The work closest to ours is [1], who provide a cost-sensitive reduction scheme for optimizing linear losses under two fairness criteria: demographic parity and equal odds. Both these criteria essentially reduce to linear constraints on the confusion matrix. Our approach can be seen as a generalization of their scheme to non-linear losses and constraints, while additionally enjoying consistency guarantees.

2 PRELIMINARIES

Notations: We denote $[n] = \{1, \ldots, n\}$, the n-dimensional probability simplex $\Delta_n = \{p \in [0,1]^n \mid \sum_{i=1}^n p_i = 1\}$, and the set of non-negative real numbers by $\mathbb{R}_+$.

We consider a general multiclass classification problem with an instance space $\mathcal{X}$ and output space $\mathcal{Y} = \{1, \ldots, n\}$. Let D denote a probability distribution over the instances and labels $\mathcal{X} \times [n]$, and $\eta_y(x) = \mathbf{P}(Y = y \,|\, X = x)$ denote the conditional class probability. Let $\pi_y = \mathbf{P}(Y = y)$

We will work with randomized classifiers $h : \mathcal{X} \to \Delta_n$ that map instances $x \in \mathcal{X}$ to a distribution over labels $h(x) \in \Delta_n$. Let $\mathcal{H}$ be the space of all randomized classifiers. We measure the performance of a classifier w.r.t. distribution D using a loss function $L : \mathcal{H} \to \mathbb{R}_+$ that associates a non-negative value $L(h; D) \in \mathbb{R}_+$ to each classifier $h \in \mathcal{H}$ (with lower values indicating better performance). We also require a classifier to satisfy a set of K inequality constraint $g_k(h; D) \leq \epsilon_k,\ k \in [K]$, where each $g_k : \mathcal{H} \to \mathbb{R}$ associates a real value to each classifier and $\epsilon_k \in \mathbb{R}$ is a threshold. Given a finite sample $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ drawn i.i.d. from D, our goal is to solve the following optimization problem:

$$\inf_{h \in \mathcal{H}} L(h; D) \quad \text{s.t.} \quad g_k(h; D) \leq \epsilon_k,\ \forall k \in [K]. \qquad \text{(OP1)}$$

We assume that there exists a classifier h^* that achieves the optimal objective value in (OP1) and satisfies all the constraints, and henceforth replace ‘inf’ with ‘min’.

Table 1: Examples of complex loss functions. For binary settings, the label space is $\{0, 1\}$.

Loss	$\psi(C)$
(Binary) F_1	$1 - \frac{2C_{11}}{2C_{11}+C_{01}+C_{10}}$
G-mean	$1 - \left(\prod_{i=1}^{n} \frac{C_{ii}}{\pi_i}\right)^{1/n}$
H-mean	$1 - n\left(\sum_{i=1}^{n} \frac{\pi_i}{C_{ii}}\right)^{-1}$
Q-mean	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{C_{ii}}{\pi_i}\right)^2}$
Micro F_1	$1 - \frac{2\sum_{i=2}^{n} C_{ii}}{2-\sum_{i=1}^{n} C_{1i}-\sum_{i=1}^{n} C_{i1}}$
Min-max	$\max_{i\in[n]}\left(1 - \frac{C_{ii}}{\pi_i}\right)$

Table 2: Examples of (convex) constraint functions.

Constraint	$\phi(C)$
(Binary) Demographic Parity	$\max_{a\in[M]}\left\|(C^a_{01} + C^a_{11}) - \frac{1}{M}\sum_{a=1}^{M}(C^a_{01} + C^a_{11})\right\|$
(Binary) Equal Odds for $y \in \{0, 1\}$	$\max_{a\in[M]}\left\|C^a_{y1} - \frac{1}{M}\sum_{a=1}^{M} C^a_{y1}\right\|$
(Binary) Coverage	$C_{01} + C_{11}$
Normalized Absolute Error	$\frac{1}{2(1-\min_y \pi_y)}\sum_{i=1}^{n}\left\|\pi_i - \sum_{j=1}^{n} C_{ji}\right\|$
KLD Error	$\sum_{i=1}^{n}\pi_i \log\left(\frac{\pi_i}{\sum_{j=1}^{n} C_{ji}}\right)$
Churn rate	$\sum_{i\neq j} C'_{i,j}$

We denote the confusion matrix for classifier h as $C^D[h] \in [0,1]^{n\times n}$, where each $C^D_{ij}[h] = \mathbf{P}(Y = i, h(X) = j)$. Here the probability is over both the draw of (X, Y) from D and the randomness in h, with $h(X) = j$ denoting the event that h predicts j on X. We will be interested in loss functions L and constraint functions g_k that can be expressed as general functions of the confusion matrix of the classifier, i.e. $L(h; D) = \psi(C^D[h])$ and $g(h; D) = \phi_k(C^D[h])$ for some $\psi : [0,1]^{n\times n}\to[0, R]$, $\phi_k : [0,1]^{n\times n}\to[0, R],\ k \in \{1, \ldots, K\}$ and $R > 0$.

Examples of loss functions of the above form include linear functions of the confusion matrix such as the classification error $\psi(C) = \sum_{i\neq j} C_{ij}$, and the cost-sensitive classification error $\psi(C) = \sum_{i,j} W_{ij}C_{ij}$ with costs $W_{ij} \in \mathbb{R}_+, \forall i, j \in [n]$. Our primary focus is on loss functions that are complex non-linear functions of the confusion matrix, examples of which are provided in Table 1. These include convex functions of the confusion matrix such as the G-mean, H-mean, and Q-mean losses popularly used in class-imbalanced classification problems [22, 30, 32], as well as, fractional-convex losses such as the F_1-measure or its multiclass variants popular in information retrieval [23]. See [16] for more examples of losses of the above form.

The following are examples of constraints on a classifier:

Example 1 (Budget/coverage constraint)**.** *The simplest constraint on a binary classifier is that the proportion of positive predictions by the classifier $C_{01} + C_{11}$ is within a budget $\epsilon \in (0, 1)$.*

Example 2 (Fairness constraints)**.** *Consider a scenario where there is an additional protected attribute $A \in [M]$ associated with each instance, and one needs to ensure that the learned classifier is fair for all values of the protected attribute. For example, in a credit risk assessment task, the protected attribute could be the race of an individual; when using a classifier to aid in college admission decisions, the protected attribute could be the gender of the applicant. Here D is a distribution over instances, labels, and values of A. Let D_a denote the distribution of (X, Y) conditioned on $A = a$ and $C^a \in [0,1]^{n\times n}$ denote the confusion matrix of a classifier w.r.t. D_a. The following are two popular notions of fairness for binary classification [20, 12]:*

Demographic parity*: A classifier is fair if the probability of positive prediction conditioned on the protected attribute A is the same for all values of A. For a small $\epsilon \in (0, 1)$, a relaxed version of this constraint requires that $\forall a \in [M]$: $\left|(C^a_{01} + C^a_{11}) - \frac{1}{M}\sum_{a=1}^{M}(C^a_{01} + C^a_{11})\right| \leq \epsilon$ [1].*

Equalized odds*: A classifier is fair if the true/false positive rate of the classifier conditioned on the protected attribute A is the same for all values of A. A relaxed version of this constraint requires that for each $a \in [M]$ and $y \in \{0, 1\}$: $\left|C^a_{y,1} - \frac{1}{M}\sum_{a=1}^{M} C^a_{y,1}\right| \leq \epsilon$, for a small $\epsilon \in (0, 1)$ [1].*

Example 3 (Quantification constraints)**.** *One version of the quantification problem seeks to learn a model that optimizes a classification loss L subject to the KL-divergence or absolute error between true class distribution $(\pi_1, \ldots, \pi_n)$ and the predicted class distribution $(\sum_i C_{i1}, \ldots, \sum_i C_{in})$ being within a limit $\epsilon > 0$.*

Example 4 (Churn constraints)**.** *Suppose there is an existing, deployed classifier $h' : \mathcal{X}\to\Delta_n$, and we wish to learn a new classifier h that optimizes a loss L subject to the churn rate (the probability of disagreement with h') being within a threshold $\epsilon \in (0, 1)$, i.e. $\mathbf{P}_X(h(X) \neq h'(X)) \leq \epsilon$. Denoting $C'_{ij} = \mathbf{P}_X(h(X) \neq h'(X)), \forall i, j$, this constraint can be equivalent written as $\sum_{i\neq j} C'_{ij} \leq \epsilon$.*

Before proceeding to algorithms for solving (OP1), we define the *empirical* confusion matrix for a classifier h on sample S as $\mathrm{conf}(h, S) \in [0,1]^{n\times n}$, where $[\mathrm{conf}(h, S)]_{ij} = \frac{1}{m}\sum_{k=1}^{m} \mathbf{1}(y_k = i, h(x_k) = j)$. We define the regret of a learning algorithm that takes sample $S \sim D^m$ and outputs classifier h_m, as the difference between the loss of h_m and that of an optimal classifier h^* to (OP1): $L(h_m) - L(h^*)$. We say the algorithm is ***statistically consistent*** if it converges in the large sample limit to the optimal loss and to zero constraint violation, i.e. for any $\nu > 0$, as $m\to\infty$, $\mathbf{P}_{S\sim D^m}\big(L(h_m) - L(h^*) > \nu\big)\to 0$ and $\mathbf{P}_{S\sim D^m}\big(g_k(h_m) - \epsilon_k > \nu\big)\to 0, \forall k \in [K]$.

3 ALGORITHMS

In this section, we describe a general framework for solving (OP1) by building on top of the ideas in Narasimhan et al. (2015) [27]. We first define the set of all confusion matrices that can be generated by some randomized classifier:

$$\mathcal{C}_D = \{C^D[h] \,|\, h \in \mathcal{H}\}.$$

The set $\mathcal{C}_D$ is convex as for any $C^D[h_1], C^D[h_2] \in \mathcal{C}_D$ and $\alpha \in (0,1)$, $\alpha C^D[h_1] + (1-\alpha)C^D[h_2] \in \mathcal{C}_D$. This confusion matrix is achieved by a classifier that predicts for any x, $h_1(x)$ with probability α and $h_2(x)$ with probability $1-\alpha$. The classification problem in (OP1) can then be viewed as a constrained optimization problem over $\mathcal{C}_D$:

$$\min_{C \in \mathcal{C}_D} \; \psi(C) \quad \text{s.t.} \quad \phi_k(C) \le \epsilon_k \; \forall k \in [K] \qquad \text{(OP2)}$$

If ψ is linear, i.e. $\psi(C) = \sum_{i,j} W_{ij}C_{ij}$ for some $W \in \mathbb{R}^{n\times n}$, and there are no constraints, the above problem reduces to a simple cost-sensitive learning problem with cost matrix W. This can be solved, for example, using a standard plug-in approach. Here we obtain an estimator $\widehat{\eta} : \mathcal{X} \to \Delta_n$ for the conditional class probability η using an algorithm like logistic regression, and construct the Bayes optimal classifier for the cost matrix W from the estimated class probabilities: $h(x) = \operatorname{argmin}_{j\in[n]} \sum_{i=1}^{n} \widehat{\eta}_i(x) W_{ij}$.

When ψ is non-linear, we can consider applying an optimization solver to (OP2) that proceeds by optimizing linear approximations to ψ, so that each subproblem can be solved using a cost-sensitive learner. This is the approach taken by [27] for unconstrained convex losses ψ, where they apply the Frank-Wolfe method [13] to solve (OP2) and formulate a sequence of cost-sensitive learning tasks. They also handle the case where ψ is an (unconstrained) ratio of linear functions of C using the bisection method [5].

Below, we extend the work of [27] to optimize convex and fractional-convex losses under constraint functions $\phi_1, \ldots, \phi_k$ that are convex in C. The classifier learned is a randomized combination of multiple plug-in classifiers (e.g. cost-weighted logistic regression classifiers).

3.1 Convex Losses with Constraints

We start with the case where ψ is convex over $\mathcal{C}_D$. Introducing Lagrange multipliers $\lambda = [\lambda_1, \ldots, \lambda_K] \in \mathbb{R}_+^K$ for the constraints, we formulate the Lagrangian for (OP2):

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

and the optimization problem in (OP2) is equivalent to:

$$\max_{\lambda \ge \mathbf{0}} \; \min_{C \in \mathcal{C}_D} \; \mathcal{L}(C, \lambda), \qquad \text{(OP3)}$$

where notice that $\mathcal{L}$ is convex over $\mathcal{C}_D$ and linear in λ.

(OP3) can be seen as a maximization of a concave objective function $F(\lambda) = \min_{C\in\mathcal{C}_D} \mathcal{L}(C, \lambda)$ in λ. We can now apply a gradient ascent procedure to maximize F over λ. This would involve computing the supergradient of G at a given λ, by performing the minimization over $C \in \mathcal{C}_D$, and evaluating the gradient $\nabla_\lambda \mathcal{L}(\widehat{C}, \lambda)$ at the minimizer $\widehat{C}$. So all we need is a way to solve the inner minimization problem. Since this is an unconstrained convex minimization over $\mathcal{C}_D$, we can make use of the Frank-Wolfe based algorithm proposed in [27] for unconstrained convex losses.

Algorithm 1 COCO: Algorithm for Convex losses with Convex Constraints

```
Input: $\psi, \phi_1, \ldots, \phi_K : [0,1]^{n\times n} \to \mathbb{R}_+$
       $S = ((x_1, y_1), \ldots, (x_m, y_m))$
Initialize: $\lambda^0 = 0^K$, $\eta_0 > 0$
For $t = 1$ to $T$ do
    $(\widehat{C}^t, \widehat{h}^t) = \texttt{FrankWolfe}(\psi, \phi_1, \ldots, \phi_K, \lambda^{t-1}, S)$
    $\lambda_k^t = \Pi_{[0,B]}\left(\lambda_k^{t-1} + \frac{\eta_0}{\sqrt{t}}(\phi_k(\widehat{C}^t) - \epsilon_k)\right)$, $\forall k$
End For
Output: Classifier $\bar{h} : \mathcal{X} \to \Delta_n$ that for any $x \in \mathcal{X}$
    outputs $\widehat{h}^t(x)$ with probability $\frac{1}{T}$
```

Algorithm 2 Frank-Wolfe based algorithm of [27]

```
1: Input: $\psi, \phi_1, \ldots, \phi_K : [0,1]^{n\times n} \to [0, R]$, $\lambda \in \mathbb{R}_+^K$
          $S = ((x_1, y_1), \ldots, (x_m, y_m))$
2: Split $S$ into $S_1$ and $S_2$ with sizes $\lceil \frac{m}{2} \rceil$ and $\lfloor \frac{m}{2} \rfloor$
3: $\Gamma^0 = \text{conf}(H^0, S_1)$ for some $H^0 : \mathcal{X} \to \Delta_n$
4: For $r = 1$ to $Q$ do
5:     $W = \nabla\psi(\Gamma^{r-1}) + \sum_{k=1}^{K} \lambda_k \nabla\phi_k(\Gamma^{r-1})$
6:     $H^r = \texttt{cost-sensitive}(W, S_2)$
7:     $\Gamma^r = \left(1 - \frac{2}{r+1}\right)\Gamma^{r-1} + \frac{2}{r+1}\text{conf}(H^r, S_1)$
8: End For
9: Output: $\widehat{C} = \Gamma^Q$, Classifier $\widehat{h} : \mathcal{X} \to \Delta_n$ that for $x \in$
   $\mathcal{X}$ outputs $H^r(x)$ with prob. $\frac{2}{r+1}\prod_{s=r+1}^{Q}\left(1 - \frac{2}{s+1}\right)$
```

The overall algorithm, which we refer to as COCO, is shown in Algorithm 1. The algorithm maintains iterates $\lambda^1, \ldots, \lambda^T \in \mathbb{R}_+^K$. In each iteration t, the algorithm invokes the Frank-Wolfe method to find a confusion matrix $\widehat{C}^t$ that (approximately) minimizes $\mathcal{L}(C, \lambda^{t-1})$, and a classifier $\widehat{h}^t$ whose (empirical) confusion matrix evaluates to $\widehat{C}^t$. This is followed by a gradient ascent update: $\lambda^t = \Pi_{[0,B]^K}\left(\lambda^t + \eta^t \nabla_\lambda \mathcal{L}(\widehat{C}^t, \lambda^{t-1})\right)$, where $\Pi_{\mathcal{A}}(\cdot)$ denotes the ℓ_2 projection onto set $\mathcal{A}$, B is a parameter that we will set later, and $\eta^t > 0$ is the step-size. The final solution after T iterations is given by $\bar{C} = \frac{1}{T}\sum_{t=1}^{T} \widehat{C}^t$. A classifier that achieves this confusion matrix $\bar{h}$ is an equal-weighted randomized combination of $\widehat{h}^1, \ldots, \widehat{h}^T$.

For completeness, we also outline the inner Frank-Wolfe (FW) method for a given λ (see Algorithm 2). In each iteration r, the FW method maintains an iterate Γ^r, computes the gradient $W = \nabla_C \mathcal{L}(\Gamma^{r-1}, \lambda)$, and minimizes a linear approximation of $\mathcal{L}(C, \lambda)$ at Γ^{r-1}: $\min_{C\in\mathcal{C}_D} \sum_{i,j} W_{ij}C_{ij}$ (lines 5–6). This is equivalent to minimizing a cost-sensitive loss with cost matrix W. The minimizing classifier H^r is used to compute the next iterate (line 7). The fi-

Algorithm 3 FRACO: Algorithm for Fractional Convex Losses with Convex Constraints

```
1: Input: $\psi(C) = \frac{f(C)}{f'(C)}$ where $f, f' : [0,1]^{n \times n} \to [0, R]$
          are $f$ is convex and $f'$ is concave in $C$
          $S = ((x_1, y_1), \ldots, (x_m, y_m))$
2: Initialize: $h^0 : \mathcal{X} \to [n]$, $\alpha^0 = 0$, $\beta^0 = R$
3: For $t = 1$ to $T$ do
4:     $\gamma^t = (\alpha^{t-1} + \beta^{t-1})/2$
5:     $(\widehat{C}^t, \widehat{h}) \leftarrow$ COCO$(f - \gamma^t f', \phi_1, \ldots, \phi_K, S)$
6:     If $\psi(\widehat{C}^t) \geq \gamma^t$ then $\alpha^t = \gamma^t, \beta^t = \beta^{t-1}, h^t = h^{t-1}$
7:                        else $\alpha^t = \alpha^{t-1}, \beta^t = \gamma^t, h^t = \widehat{h}$
8: End For
9: Output: $\bar{h} = h^T : \mathcal{X} \to \Delta_n$
```

nal classifier is a randomized combination of $H^1, \ldots, H^Q$. Due to a technical requirement for our theoretical results (see Section 4), we use separate samples for the learning step in line 6, and the evaluation steps in lines 3 and 7.

We perform cost-sensitive learning using the plug-in approach mentioned previously, where we suitably weight a class probability estimator $\widehat{\eta}$ such as logistic regression. Note that $\widehat{\eta}$ needs to be learned only once and can be re-used in each invocation of the cost-sensitive learner

It is worth mentioning that the COCO method is a generalization of the fair classification technique of [1] for linear losses and constraints. The authors similarly formulate a Lagrangian for the constrained classification problem and observe that for any assignment of the Lagrange multipliers, the inner minimization can be solved using a cost-sensitive learner. On the other hand, our approach handles general convex losses and constraints by nesting a gradient ascent procedure together with the Frank-Wolfe method, and additionally comes with consistency guarantees.

3.2 Fractional-convex Losses with Constraints

The second family of constrained classification problems that we consider involves fractional-convex losses $\psi(C) = \frac{f(C)}{f'(C)}$, where f is convex and f' is concave over $\mathcal{C}_D$. Examples include the F_1 measure loss and the micro F_1 measure loss shown in Table 1. In this case, we prescribe a variant of the Bisection method proposed in [27] for unconstrained fractional-linear losses. This algorithm is based on the following simple observation: if C^* is an optimal solution to (OP2), then checking whether $\psi(C^*) \geq \gamma$ is equivalent to checking if the inequality $f(C) - \gamma f'(C) \geq 0$ holds for all $C \in \mathcal{C}_D$ that satisfy $\phi_k(C) \leq \epsilon_k, \forall k \in [K]$. Thus to solve (OP2), it is enough to solve the following simpler optimization problem for different values of $\gamma \in [0, R]$, and pick the largest γ for which the objective value is non-negative:

$$\min_{C \in \mathcal{C}_D} \; f(C) - \gamma f'(C) \;\; \text{s.t.} \;\; \phi_k(C) \leq \epsilon_k, \forall k \in [K]. \quad (2)$$

Notice that this is a convex minimization problem over $\mathcal{C}_D$ with convex constraint functions. One can therefore apply the COCO algorithm described in the previous section to perform this optimization. The overall method, referred to as FRACO, is outlined in Algorithm 3. Rather than performing a linear search on γ, the algorithm performs a more efficient binary search: at each iteration t, the algorithm maintains an upper bound $\beta^t \leq R$ and a lower bound $\alpha^t \geq 0$ on $\psi(C^*)$, uses the COCO method to check if (2) yields a non-negative value for $\gamma^t = (\alpha^t + \beta^t)/2$, if true raises the lower bound to $\alpha^t = \gamma^t$, and if false, lowers the upper bound to $\beta^t = \gamma^t$. The algorithm outputs a classifier h^T with loss between α^T and β^T, and which satisfies the constraints.

3.3 Extension to Fairness Constraints

In the special case of fairness constraints, there is a protected attribute $A \in [M]$ associated with each instance, and a classifier $h : \mathcal{X} \times [M] \to \Delta_n$ maps a given instance and its protected attribute (x, a) to $h(x, a) \in \Delta_n$. Recall that here D is a distribution over instances, labels and the values of A, and we will be interested in the conditional distributions $D_1, \ldots, D_M$, and the conditional class probabilities $\eta_y(x, a) = \mathbf{P}(Y = y \,|\, X = x, A = a), \forall y \in [n]$. The loss function takes the form $L(h) = \psi(C^{D_1}[h], \ldots, C^{D_M}[h])$ for some $\psi : ([0,1]^{n \times n})^M \to [0, R]$, with constraint functions $g_k(h) = \phi_k(C^{D_1}[h], \ldots, C^{D_M}[h])$ for some $\phi_k : ([0,1]^{n \times n})^M \to [0, R]$, $\forall k \in [K]$. We define the space of confusion matrices $\mathcal{C}_D = \{(C^{D_1}[h], \ldots, C^{D_M}[h]) \,|\, h \in \mathcal{H}\}$ and the optimization problem in (OP2) becomes:

$$\min_{(C^1, \ldots, C^M) \in \mathcal{C}_D} \psi(C^1, \ldots, C^M) \qquad \text{(OP4)}$$
$$\text{s.t.} \;\; \phi_k(C^1, \ldots, C^M) \leq \epsilon_k \;\; \forall k \in [K]$$

Given a sample $S = \{(x_1, y_1, a_1), \ldots, (x_m, y_m, a_m)\}$ of instances, labels and protected attribute values, the COCO and FRACO methods readily apply to this setting. We outline the variant of COCO for fairness constraints in Algorithm 4 in the appendix and highlight the changes below.

Firstly, in the place of confusion matrix $\widehat{C}^t$ maintained in each iteration, we will have a set of M confusion matrices $\widehat{C}^{1,t}, \ldots \widehat{C}^{M,t}$. Secondly, the Frank-Wolfe solver will maintain M iterates $\Gamma^{1,r}, \ldots, \Gamma^{M,r}$ and M cost-matrices $W^1, \ldots, W^M \in \mathbb{R}_+^{n \times n}$, one for each protected group. Thirdly, the cost-sensitive learner in the algorithm takes M cost matrices $W^1, \ldots, W^M$ as input and seeks to minimize the loss $\sum_{a=1}^{M} \sum_{i,j} W_{ij}^a C_{ij}^a$. The plug-in classifier that optimizes this loss uses an estimator $\widehat{\eta} : \mathcal{X} \times [M] \to \Delta_n$ for the conditional class probabilities, and takes the form: $h(x, a) = \operatorname{argmin}_{j \in [n]} \sum_{i=1}^{n} \widehat{\eta}_i(x, a) W_{ij}^a$.

We close by noting that all our methods apply to a general n-class problem, with run-time polynomial in n. In contrast, standard methods for complex losses such as SVM-Perf [14] or plug-in methods [21] do not extend tractably to multiclass problems as they need to perform a search over a parameter space of size exponential in n.

4 CONSISTENCY GUARANTEES

We now provide regret bound guarantees for the algorithms developed in the previous section. We will assume (OP1) is feasible and use h^* to denote an optimal solution to the problem. The proofs are provided in the appendix.

Convex Losses with Constraints. We first derive a regret bound for the COCO algorithm in terms of the regret of the inner Frank-Wolfe algorithm. Our result applies to settings where strong duality holds for (OP3). Indeed, this is the case for all the constraint functions in Table 2 (e.g. they satisfy Slater's conditions for strong duality [5]). Let $\lambda^* \in \mathbb{R}_+^K$ be the optimal Lagrange multipliers for (OP3). We also assume that $\psi, \phi_1, \ldots \phi_k : [0,1]^{n\times n} \rightarrow [0, R]$ are G-Lipschitz w.r.t. the ℓ_1 norm.

Theorem 1 (Regret Bound for COCO)**.** *For any $\delta \in (0,1]$, let the following hold w.p. $\geq 1-\delta$ (over $S \sim D^m$): for each iteration $t \in [T]$, the Frank-Wolfe algorithm satisfies $\mathcal{L}(\widehat{C}^t, \lambda^t) \leq \min_{C\in\mathcal{C}_D} \mathcal{L}(C, \lambda^t) + \theta(\delta, m)$, and $\|C^D[\widehat{h}^t] - \widehat{C}^t\|_1 \leq \xi(\delta, m)$, where $\theta, \xi : (0,1] \times \mathbb{N} \rightarrow \mathbb{R}_+$. Let parameter B be s.t. $B \geq 2\max_{k\in[K]} \lambda_k^*$. Let $\bar{h} : \mathcal{X} \rightarrow \Delta_n$ be the classifier obtained after $T = \tau m$ iterations, for some $\tau \in \mathbb{N}$. Then w.p. $\geq 1-\delta$ (over $S \sim D^m$):*

$$L(\bar{h}) \leq L(h^*) + \theta(\delta, m) + G\xi(\delta, m) + \widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{m}}\right).$$

and $\forall k \in [K]$,

$$g_k(\bar{h}) \leq \epsilon_k + \frac{2}{B}\theta(\delta, m) + G\xi(\delta, m) + \widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{m}}\right).$$

The regret of the Frank-Wolfe algorithm can be further bounded using a result from [27].

Theorem 2 (Regret Bound for Frank-Wolfe Algorithm [27])**.** *Let $\psi, \phi_1, \ldots, \phi_K : [0,1]^{n\times n} \rightarrow [0, R]$ be β-smooth w.r.t. the ℓ_1 norm. Let $\widehat{\eta} : \mathcal{X} \rightarrow \Delta_n$ be the conditional class probability model used to construct the plug-in classifier for the cost-sensitive learner in the Frank-Wolfe algorithm. For a given $\lambda \in [0, B]^K$, let $(\widehat{C}, \widehat{h})$ be returned by the algorithm after $Q = \kappa m$ iterations, for some $\kappa \in \mathbb{N}$. Then for any $\delta \in (0,1]$, w.p. $\geq 1-\delta$ (over $S \sim D^m$)*

$$\begin{aligned} &\mathcal{L}(\widehat{C}, \lambda) - \min_{C\in\mathcal{C}_D} \mathcal{L}(C, \lambda) \\ &\quad\leq G\rho\mathbf{E}_X\left[\left\|\widehat{\eta}(X) - \eta(X)\right\|_1\right] + \frac{2\beta\rho}{\kappa m + 2} + \widetilde{\mathcal{O}}\left(\frac{n^3}{\sqrt{m}}\right) \end{aligned}$$

$$\text{and } \|C^D[\widehat{h}] - \widehat{C}\|_1 \leq \widetilde{\mathcal{O}}\left(\frac{n^3}{\sqrt{m}}\right),$$

where $\rho = 4(1 + KB)$.

This theorem requires the loss and constraint functions to be strongly smooth. If this is not true (e.g. with the G-mean or H-mean), we can apply the Frank-Wolfe algorithm to a suitable smooth approximation to the loss/constraint functions and derive a similar guarantee; see [27] for details.[1]

The above bound depends on the quality of the class probability estimates $\widehat{\eta}$ used by the plug-in classifier, as captured by the term $\mathbf{E}_X[\|\widehat{\boldsymbol{\eta}}(X) - \boldsymbol{\eta}(X)\|_1]$.[2] Combining Theorems 1 and 2, we get that as sample size $m \rightarrow \infty$, both $\epsilon(\delta, m) \rightarrow 0$ and $\xi(\delta, m) \rightarrow 0$. Suppose in addition, the class probability estimation algorithm we use is such that $\mathbf{E}_X[\|\widehat{\boldsymbol{\eta}}(X) - \boldsymbol{\eta}(X)\|_1] \rightarrow 0$ as $m \rightarrow \infty$, i.e. the estimation error goes to zero with increasing sample size.[3] Then when $m \rightarrow \infty$, COCO converges to the optimal loss and to zero constraint violation, and is *statistically consistent*.

Recall that in the Frank-Wolfe algorithm, we use separate samples for the cost-sensitive learning (line 6), and for estimating the confusion matrix of the learned classifier (line 7). This is a technical requirement for proving Theorem 2 that ensures that the samples used for the training and evaluation steps within the algorithm are independent and identically distributed.[4]

As another technical requirement, the projection parameter B in the Frank-Wolfe algorithm needs to be set to a value $\geq 2\max_{k\in[K]} \lambda_k^*$. This is required to ensure convergence of the outer gradient ascent solver. In our experiments, we find it sufficient to set B to a large value.

Fractional-convex Losses with Constraints. We bound the regret for the FRACO in terms of the regret for the COCO algorithm, invoked in each iteration. Assume $f, f', \phi_1, \ldots, \phi_k$ are G-Lipschitz w.r.t. the ℓ_1 norm.

Theorem 3 (Regret Bound for FRACO)**.** *Let $f'(C) \geq b, \forall C \in \mathcal{C}_D$, for $b > 0$. Let for any $\delta \in (0,1]$, w.p. $\geq 1-\delta$, in each iteration t, COCO satisfies: $f(\widehat{C}^t) - \gamma^t f'(\widehat{C}^t) \leq \min_{C\in\mathcal{C}_D} f(C) - \gamma^t f'(C) + \theta(\delta, m)$, with each $\phi_k(\widehat{C}^t) \leq \epsilon_k + \theta'(\delta, m)$, and $\|C^D[\widehat{h}] - \widehat{C}^t\|_1 \leq \xi(\delta, m)$. Let $\bar{h}$ be the classifier returned after $T = \tau m$ iterations, for some $\tau \in \mathbb{N}$. Then $\forall\delta \in (0,1]$, w.p. $\geq 1-\delta$ (over $S \sim D^m$)*

$$\begin{aligned} L(\bar{h}) &\leq L(h^*) + \frac{1}{b}\theta(\delta, m) + \frac{(R+1)G}{b}\xi(\delta, m) + 2^{-\tau m}R \\ &\textit{and} \quad g_k(\bar{h}) \leq \epsilon_k + \theta'(\delta, m),\ \forall k \in [K]. \end{aligned}$$

One can verify that the conditions required in the above result are satisfied by both the F_1-measure and the micro F_1 loss functions. Clearly, as long as the COCO routine is statistically consistent, i.e. $\theta(\delta, m) \rightarrow 0$, $\theta'(\delta, m) \rightarrow 0$ and $\xi(\delta, m) \rightarrow 0$ as $m \rightarrow \infty$, so is the FRACO algorithm.

[1] The KLD error can be unbounded when $\sum_j C_{ji} = 0$ for any $i \in [n]$. For our theorems to hold, we can work with a smoothed variant, e.g. $\sum_i \pi_i \log\left(\frac{\pi_i}{\sum_j C_{ji} + \nu}\right)$, for a small $\nu > 0$.

[2] For example, this estimation error can be high when the conditional class probabilities are not well-calibrated.

[3] This includes algorithms that minimize a strictly proper composite loss (e.g. logistic) over a sufficiently rich function class [31]

[4] In our experiments, to make the best use of the available data, we use the same sample in both steps.

Table 3: Datasets used in experiments.

Dataset	# instances	# features	# classes	p.attr
adult	32561	106	2	gender
compas	4020	20	2	race
crimes	1993	198	2	race
default	30000	24	2	gender
pageblocks	5473	10	5	-
abalone	4177	10	12	-

Dataset	ϵ	COCO	FW-unc	LogReg
adult	0.25	0.23 (0.244)	0.18 (0.345)	0.27 (0.196)
compas	0.25	0.38 (0.242)	0.31 (0.385)	0.67 (0.069)
crimes	0.25	0.27 (0.250)	0.16 (0.397)	0.18 (0.331)
default	0.25	0.40 (0.242)	0.34 (0.405)	0.61 (0.075)

Table 4: Minimizing H-mean loss s.t. coverage $\leq \epsilon$. The test H-mean loss is shown for each method and the proportion of positive predictions is provided in parenthesis. A constant classifier would receive a H-mean loss of 1.

Consistency under Fairness Constraints. The above guarantees easily extend to constraints for fairness. Below, we present a regret bound result for the COCO algorithm with fairness constraints. Let $h^* : \mathcal{X} \times [M] \rightarrow [n]$ denote the optimal solution to (OP4), and $\lambda^* \in \mathbb{R}_+^K$ denote optimal Lagrange multipliers. Assume that $\psi, \phi_1, \ldots \phi_k : ([0,1]^{n \times n})^M \rightarrow [0, R]$ are G-Lipschitz w.r.t. the ℓ_1 norm.

Theorem 4 (Regret Bound for COCO with fairness constraints)**.** *For any $\delta \in (0,1]$, let the following hold w.p. $\geq 1 - \delta$: in each iteration $t \in [T]$, the Frank-Wolfe algorithm satisfies: $\mathcal{L}(C^{1,t}, \ldots, C^{M,t}, \lambda^t) \leq \min_{(C^1,\ldots,C^M) \in \mathcal{C}_D} \mathcal{L}(C^1, \ldots, C^M, \lambda^t) + \theta(\delta, m)$, with $\|C^{D_a}[\widehat{h}^t] - \widehat{C}^{a,t}\|_1 \leq \xi(\delta, m), \forall a \in [M]$. Let $B \geq 2\max_{k \in [K]} \lambda_k^*$. Let $\bar{h} : \mathcal{X} \times [M] \rightarrow \Delta_n$ be the classifier returned by the algorithm after $T = \tau m$ iterations, for some $\tau \in \mathbb{N}$. Then w.p. $\geq 1 - \delta$ (over $S \sim D^m$)*

$$L(\bar{h}) \leq L(h^*) + \theta(\delta, m) + GM\xi(\delta, m) + \widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{m}}\right).$$

and $\forall k \in [K]$,

$$g_k(\bar{h}) \leq \epsilon_k + \frac{2}{B}\theta(\delta, m) + GM\xi(\delta, m) + \widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{m}}\right).$$

We show in Appendix A.3 that under the smoothness conditions in Theorem 2, $\epsilon(\delta, m) \rightarrow 0$ and $\xi(\delta, m) \rightarrow 0$ as $m \rightarrow \infty$. This together with Theorem 4 implies that the COCO algorithm is consistent under fairness constraints. To the best of our knowledge, this the first work to study the statistical consistency of algorithms for fair classification.

5 EXPERIMENTS

We ran experiments on four benchmark datasets for fair classification: (i) adult: the task is to predict if a person's income is greater than \$50K/yr, with gender as the protected attribute [9]; (ii) compas: the task is to predict whether a convicted person would commit a crime in the next two years, with race as the protected attribute [2]; (iii) crimes: the task is to predict if the crime rate in a given region in the US is above the average, with the majority race as the protected attribute [9]; (iv) default: the task is to predict if a credit card user would default on a payment, with gender as the protected attribute [9]. We also included two multiclass UCI datasets [9] (see Table 3).

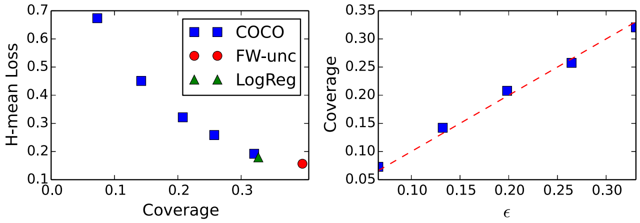


Figure 1: Plot of H-mean and coverage values achieved by COCO for different ϵ (left) and plot of coverage as a function of ϵ (right), for the ***crimes*** dataset.

We used $2/3$-rd of the data for training and $1/3$-rd for testing, averaging the results over 5 random train-test splits. The cost-sensitive learner in the Frank-Wolfe algorithm was implemented using the plug-in approach. We used linear logistic regression to estimate $\widehat{\eta}$, with the protected attribute included as one of the features.[5],[6]

Convex Losses with Constraints

Coverage. We begin with the H-mean loss subject to a coverage constraint (the proportion of positive predictions be within a given ϵ). For comparison, we included the Frank-Wolfe method (FW-unc) [27] for optimizing H-mean without constraints, and plain (unweighted) logistic regression that optimizes the 0-1 error. The results for the four binary datasets are summarized in Table 4, with the coverage values provided in parenthesis. The average run-time of COCO across train-test splits varied from 20 to 200 secs (e.g. 19.1 sec on the crimes data and 202.2 secs on the adult data).

COCO yields very small constraint violations, with significantly lower losses than LogReg. As expected, the unconstrained FW method yields the lowest loss, but incurs large constraint violations. This is further confirmed by the left side plots in Figure 1, which show loss and coverage values for different values of ϵ. Smaller values of ϵ lead to larger losses. On the right side, we have plots of coverage as a function of ϵ. Clearly, the proposed method closely satisfies the coverage constraint.

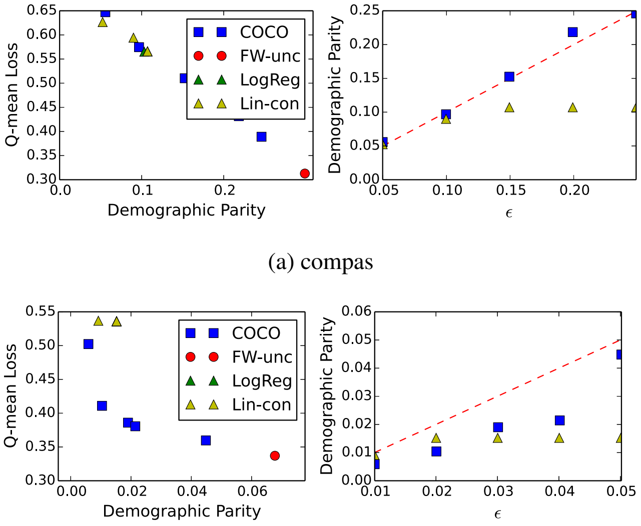
Fairness. We next considered the task of fair classification. Given that the datasets considered have imbalanced classes, Q-mean could be a loss function of interest here. We op-

[5] COCO was run for 100 iterations and the inner FW algorithm was run for 10 iterations, resulting in a total of 1000 plug-in classifiers in the final model. FRACO was run till $|\alpha^t - \beta^t| \leq 0.01$. The parameter η_0 in COCO was chosen from $\{10^{-2}, \ldots, 10^3\}$ to minimize $\max_k |g_k(\bar{h}) - \epsilon_k|$ on the training set.

[6] Code for the algorithms is available at `https://github.com/hnarasimhan/constrained-classification`

Dataset	ϵ	COCO	FW-unc	Lin-con
adult	0.05	0.33 (0.035)	0.19 (0.152)	0.39 (0.027)
compas	0.20	0.41 (0.206)	0.33 (0.279)	0.57 (0.107)
crimes	0.20	0.32 (0.197)	0.16 (0.436)	0.52 (0.190)
default	0.05	0.37 (0.032)	0.35 (0.060)	0.54 (0.015)

Table 5: Optimizing Q-mean s.t. Demographic Parity $\leq \epsilon$. The Q-mean loss and DP values (in parenthesis) are reported. A constant classifier would receive a loss of 0.71.



(a) compas

(b) default

Figure 2: Optimizing Q-mean s.t. Demographic Parity $\leq \epsilon$.

timized the Q-mean loss subject to the demographic parity (DP) being within a given ϵ. In this case, we additionally compare with a classifier that optimizes (linear) classification loss subject to the DP constraint (Lin-con). This baseline is a representative of the kind of formulations adopted by existing fair classification algorithms (e.g. [1]).

Our results are shown in Table 5 and Figure 2. COCO often yields significantly lower loss than Lin-con. While Lin-con satisfies the constraints, it tapers at a certain value of ϵ and is unable to improve on the Q-mean even when higher constraint violations are allowed. This shows the benefit of directly optimizing the Q-mean instead of the 0-1 error.

Quantification. We also considered a multiclass quantification task. The goal was to minimize the Q-mean loss subject to the normalized absolute error (NAE), being within a given ϵ. NAE measures a classifier's ability to predict the prevalence of each class accurately. As seen in Figure 3, COCO incurs small constraint violations, and yields Q-mean values between that of FW-unc and LogReg.

Fractional-convex Losses with Constraints

Continuing with quantification tasks, we now seek to optimize the F_1 measure loss subject to the KLD error being within ϵ. The loss is fractional-convex, while the constraint function is convex. We compared the FRACO method with the Bisection based algorithm (BS-unc) in [27] for unconstrained optimization of the F_1 measure, and plain logistic regression. As seen in Table 6, FRACO satisfies the constraint for all datasets except compas, where there is a mild violation. In terms of the loss, FRACO often performs comparable to BS-unc, and better than LogReg.

Dataset	ϵ	FRACO	BS-unc	LogReg
adult	0.001	0.31 (0.001)	0.31 (0.003)	0.34 (0.007)
compas	0.001	0.64 (0.003)	0.51 (0.020)	0.70 (0.105)
crimes	0.001	0.21 (0.001)	0.21 (0.002)	0.22 (0.002)
default	0.001	0.49 (0.001)	0.50 (0.005)	0.64 (0.107)

Table 6: Optimizing F-measure loss s.t. KLD error $\leq \epsilon$.

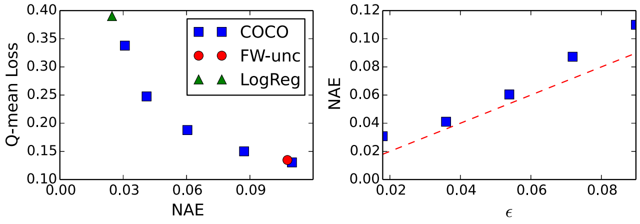


Figure 3: Optimizing (multiclass) Q-mean loss s.t. NAE $\leq \epsilon$, for the ***pageblocks*** dataset.

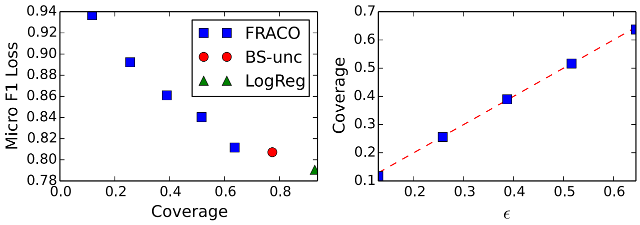


Figure 4: Optimizing (multiclass) micro F_1 loss s.t. coverage of all classes except 1 is within ϵ, for ***abalone***.

Our final experiment was on the multiclass micro F_1 loss. The version presented in Table 1 treats class 1 as a default class (e.g. the non-relevant class in a retrieval task), and evaluates performance relative to this class. We constrained the proportion of all classes other than class 1 to be within a desired coverage value ϵ. The results shown in Figure 4 once again confirm the effectiveness of our method.

6 CONCLUSION

We have developed algorithms for optimizing complex loss functions under complex constraints, and demonstrated their utility on a variety of problems. In the future, it would be interesting to explore ways to reduce the number of base classifiers needed to construct the final classifier in our algorithms, and to scale our approach to large settings.

Acknowledgments. HN thanks Harish G. Ramaswamy for helpful discussions, and Sanjay Chawla, Purushottam Kar and Fabrizio Sebastiani for introducing him to the quantification problem. HN also thanks the Institute for Applied Computational Science, Harvard University for funding his travel to the conference.

References

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

[2] A. Barry-Jester, B. Casselman, and D. Goldstein. The new science of sentencing. The Marshall Project, 2015.

[3] Y. Bechavod and K. Ligett. Learning fair classifiers: A regularization-inspired approach. In *FAT/ML*, 2017.

[4] R. Berk, H. Heidari, S. Jabbari, M. Joseph, M. Kearns, J. Morgenstern, S. Neel, and A. Roth. A convex framework for fair regression. *CoRR*, abs/1706.02409, 2017.

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

[6] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq. Algorithmic decision making and the cost of fairness. In *KDD*, 2017.

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

[8] G. Forman. Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17(2):164–206, 2008.

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

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

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

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

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

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

[15] T. Kamishima, S. Akaho, and J. Sakuma. Fairness-aware learning through regularization approach. In *ICDMW*, 2011.

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

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

[18] P. Kar, H. Narasimhan, and P. Jain. Surrogate functions for maximizing precision at the top. In *ICML*, 2015.

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

[20] J. Kleinberg, S. Mullainathan, and M. Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016.

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

[22] 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. 1998.

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

[24] M. Mahdavi, T. Yang, and R. Jin. Stochastic convex optimization with multiple objectives. In *NIPS*, 2013.

[25] A. K. Menon and R. C. Williamson. The cost of fairness in classification. *CoRR*, abs/1705.09055, 2017.

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

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

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

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

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

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

[32] 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.

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

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