

---

# Consistent Multiclass Algorithms for Complex Performance Measures

---

Harikrishna Narasimhan\*

Harish G. Ramaswamy\*

Aadirupa Saha

Shivani Agarwal

Indian Institute of Science, Bangalore 560012, INDIA

HARIKRISHNA@CSA.IISC.ERNET.IN

HARISH\_GURUP@CSA.IISC.ERNET.IN

AADIRUPA.SAHA@CSA.IISC.ERNET.IN

SHIVANI@CSA.IISC.ERNET.IN

## Abstract

This paper presents new consistent algorithms for multiclass learning with complex performance measures, defined by arbitrary functions of the confusion matrix. This setting includes as a special case all loss-based performance measures, which are simply linear functions of the confusion matrix, but also includes more complex performance measures such as the multiclass G-mean and micro  $F_1$  measures. We give a general framework for designing consistent algorithms for such performance measures by viewing the learning problem as an optimization problem over the set of feasible confusion matrices, and give two specific instantiations based on the Frank-Wolfe method for concave performance measures and on the bisection method for ratio-of-linear performance measures. The resulting algorithms are provably consistent and outperform a multiclass version of the state-of-the-art SVMperf method in experiments; for large multiclass problems, the algorithms are also orders of magnitude faster than SVMperf.

## 1. Introduction

In many practical applications of machine learning, the performance measure used to evaluate the performance of a classifier takes a complex form, and is not simply the expectation or sum of a loss on individual examples. Indeed, this is the case with the G-mean, H-mean and Q-mean performance measures used in class imbalance settings (Sun et al., 2006; Wang & Yao, 2012; Kennedy et al., 2009; Kim et al., 2013; Lawrence et al., 1998), the micro and macro  $F_1$  measures used in information retrieval (IR) applications

(Lewis, 1991), the min-max measure used in detection theory (Vincent, 1994), and many others. Unlike loss-based performance measures, which are simply linear functions of the confusion matrix of a classifier, these complex performance measures are defined by general functions of the confusion matrix. How can we design consistent learning algorithms for such complex performance measures?

While there has been much interest in designing consistent algorithms for various types of supervised learning problems in recent years, most of this work has focused on loss-based performance measures, including binary/multiclass 0-1 loss (Bartlett et al., 2006; Zhang, 2004a;b; Lee et al., 2004; Tewari & Bartlett, 2007), losses for specific problems such as multilabel classification (Gao & Zhou, 2011) and ranking (Cossock & Zhang, 2008; Xia et al., 2008; Duchi et al., 2010; Ravikumar et al., 2011; Buffoni et al., 2011; Calauzènes et al., 2012), and some work on general multiclass losses (Steinwart, 2007; Ramaswamy & Agarwal, 2012; Pires et al., 2013; Ramaswamy et al., 2013).

There has also been much interest in designing algorithms for more complex performance measures. A prominent example is the SVM<sup>perf</sup> algorithm (Joachims, 2005), which was developed primarily for the binary setting; other examples include algorithms for the binary  $F_1$ -measure and its multiclass and multilabel variants (Musicant et al., 2003; Ye et al., 2012; Dembczynski et al., 2011; 2013; Parambath et al., 2014). More recently, there has been increasing interest in designing *consistent* algorithms for complex performance measures; however, most of this work has focused on the binary case (Ye et al., 2012; Menon et al., 2013; Koyejo et al., 2014; Narasimhan et al., 2014).

In this paper, we develop a general framework for designing provably consistent algorithms for complex multiclass performance measures. Our approach involves viewing the learning problem as an optimization problem over the set of feasible confusion matrices, and solving (approximately, based on the training sample) this optimization problem using an optimization method that needs access to only an approximate linear minimization routine and a sample-

---

\*Both authors made equal contributions to the paper.

*Proceedings of the 32<sup>nd</sup> International Conference on Machine Learning*, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

based confusion matrix calculator. We give two specific instantiations based on the Frank-Wolfe method for concave performance measures (such as the multiclass G-mean, H-mean and Q-mean) and on the bisection method for ratio-of-linear performance measures (such as the micro  $F_1$ ). The resulting algorithms are provably consistent, and outperform a multiclass version of SVM<sup>perf</sup> both in terms of generalization performance and in terms of training time.

**Notation.** For  $n \in \mathbb{Z}_+$ , we denote  $[n] = \{1, \dots, n\}$  and  $\Delta_n = \{\mathbf{p} \in \mathbb{R}_+^n : \sum_{i=1}^n p_i = 1\}$ . For  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ , we denote  $\|\mathbf{A}\|_1 = \sum_{i,j} |A_{ij}|$  and  $\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{i,j} A_{i,j} B_{i,j}$ . The notation  $\operatorname{argmin}_{i \in [n]}$  will denote ties being broken in favor of the larger number.

## 2. Complex Performance Measures

We are interested in general multiclass learning problems with instance space  $\mathcal{X}$  and label space  $\mathcal{Y} = [n]$ . Given a finite training sample  $S = ((x_1, y_1), \dots, (x_m, y_m)) \in (\mathcal{X} \times [n])^m$ , the goal is to learn a multiclass classifier  $h_S : \mathcal{X} \rightarrow [n]$ , or more generally, a *randomized* multiclass classifier  $h_S : \mathcal{X} \rightarrow \Delta_n$  (which given an instance  $x$  predicts a class label in  $[n]$  according to the probability distribution specified by  $h_S(x)$ ). We assume examples are drawn iid from some distribution  $D$  on  $\mathcal{X} \times [n]$ , with marginal  $\mu$  on  $\mathcal{X}$ ,  $\eta_i(x) = \mathbf{P}(Y = i | X = x)$ , and  $\pi_i = \mathbf{P}(Y = i)$ .

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

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

where the probability is over the draw of  $(X, Y)$  from  $D$  when  $h$  is deterministic, and additionally over the randomness in  $h$  when  $h$  is randomized. Clearly,  $\sum_{i,j} C_{ij}^D[h] = 1$ .

We will be interested in general, complex performance measures that can be expressed as an arbitrary function of the entries of the confusion matrix  $\mathbf{C}^D[h]$  (see Figure 1).

**Definition 2** (Performance measure). *For any function  $\psi : [0, 1]^{n \times n} \rightarrow \mathbb{R}_+$ , define the  $\psi$ -performance measure of  $h$  w.r.t.  $D$  as follows (we will adopt the convention that higher values of  $\psi$  correspond to better performance):*

$$\mathcal{P}_D^\psi[h] = \psi(\mathbf{C}^D[h]).$$

As the following examples show, this formulation captures both common loss-based performance measures, which are effectively linear functions of the entries of the confusion matrix, and more complex performance measures such as the G-mean, micro  $F_1$ -measure, and several others.

**Example 1** (Loss-based performance measures). *Consider a multiclass loss matrix  $\mathbf{L} \in [0, 1]^{n \times n}$ , such that  $L_{ij}$  represents the loss incurred on predicting class  $j$  when the true*

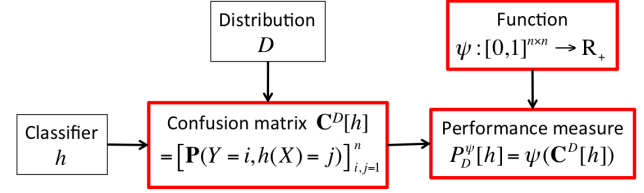


Figure 1. Complex multiclass performance measures, given by arbitrary functions of the confusion matrix, generalize both common loss-based performance measures, and binary performance measures expressed in terms of TP, TN, FP and FN. (In practice, the distribution  $D$  is unknown; one estimates the confusion matrix from a finite sample, and applies  $\psi$  to the estimated matrix.)

class is  $i$  (note that one can always shift and scale a loss matrix so that its entries lie in  $[0, 1]$  without impacting the learning problem). In such settings, the performance of a classifier  $h$  is measured by the expected loss on a new example from  $D$ , which amounts to taking a linear function of the confusion matrix  $\mathbf{C}^D[h]$ :

$$\begin{aligned} \mathcal{P}_D^{\mathbf{L}}[h] &= \mathbf{E}[1 - L_{Y, h(X)}] \\ &= \sum_{i,j} (1 - L_{ij}) C_{ij}^D[h] = \psi^{\mathbf{L}}(\mathbf{C}^D[h]), \end{aligned}$$

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

**Example 2** (Binary performance measures). *In the binary setting, where  $n = 2$  and the labels are often indexed as  $\mathcal{Y} = \{-1, 1\}$ , the confusion matrix of a classifier contains the proportions of true negatives ( $C_{-1,-1} = \text{TN}$ ), false positives ( $C_{-1,1} = \text{FP}$ ), false negatives ( $C_{1,-1} = \text{FN}$ ), and true positives ( $C_{1,1} = \text{TP}$ ). Our framework therefore includes any binary performance measure that is expressed as a function of these quantities, including the ‘balanced accuracy’ or AM measure (Menon et al., 2013) given by  $\psi^{\text{AM}}(\mathbf{C}) = \frac{1}{2}(\frac{\text{TP}}{\text{TP}+\text{FN}} + \frac{\text{TN}}{\text{TN}+\text{FP}})$ , the  $F_\beta$ -measure ( $\beta > 0$ ) given by  $\psi^{F_\beta}(\mathbf{C}) = \frac{(1+\beta^2)\text{TP}}{(1+\beta^2)\text{TP}+\beta^2\text{FN}+\text{FP}}$ , all ‘ratio-of-linear’ binary performance measures (Koyejo et al., 2014), and more generally, all ‘non-decomposable’ binary performance measures (Narasimhan et al., 2014).<sup>1</sup>*

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

$$\psi^{\text{GM}}(\mathbf{C}) = \left( \prod_{i=1}^n \frac{C_{ii}}{\sum_{j=1}^n C_{ij}} \right)^{1/n}.$$

<sup>1</sup>The ‘non-decomposable’ performance measures considered by Narasimhan et al. (2014) were expressed as functions of  $\text{TPR} = \frac{\text{TP}}{\text{TP}+\text{FN}}$ ,  $\text{TNR} = \frac{\text{TN}}{\text{TN}+\text{FP}}$ , and  $p = \text{TP} + \text{FN}$ .

Table 1. Examples of complex multiclass performance measures.

Performance measure	$\psi(\mathbf{C})$
G-mean	$(\prod_{i=1}^n \frac{C_{ii}}{\sum_{j=1}^n C_{ij}})^{1/n}$
H-mean	$n(\sum_{i=1}^n \frac{\sum_{j=1}^n C_{ij}}{C_{ii}})^{-1}$
Q-mean	$1 - \sqrt{\frac{1}{n} \sum_{i=1}^n (1 - \frac{C_{ii}}{\sum_{j=1}^n C_{ij}})^2}$
Micro $F_1$	$\frac{2 \sum_{i=2}^n C_{ii}}{2 - \sum_{i=1}^n C_{1i} - \sum_{i=1}^n C_{i1}}$
Macro $F_1$	$\frac{1}{n} \sum_{i=1}^n \frac{2C_{ii}}{\sum_{j=1}^n C_{ij} + \sum_{j=1}^n C_{ji}}$
Spectral norm	$\ \mathbf{C}^\circ\ _*$ (where $\mathbf{C}^\circ$ is obtained from $\mathbf{C}$ by normalizing rows to sum to 1 and setting diagonal entries to 0)
Min-max	$\min_{i \in [n]} \frac{C_{ii}}{\sum_{j=1}^n C_{ij}}$

**Example 4** (Micro  $F_1$ -measure). *The micro  $F_1$ -measure is widely used to evaluate multiclass classifiers in information retrieval and information extraction applications (Manning et al., 2008). Many variants have been studied; we consider here the form used in the BioNLP challenge (Kim et al., 2013), which treats class 1 as a ‘default’ class and is effectively given by the function<sup>2</sup>*

$$\psi^{\text{micro } F_1}(\mathbf{C}) = \frac{2 \sum_{i=2}^n C_{ii}}{2 - \sum_{i=1}^n C_{1i} - \sum_{i=1}^n C_{i1}}.$$

Other examples of performance measures that are given by (complex) functions of the confusion matrix include the macro  $F_1$ -measure (Lewis, 1991), the H-mean (Kennedy et al., 2009), the Q-mean (Lawrence et al., 1998), the spectral norm measure (Ralaivola, 2012; Machart & Ralaivola, 2012; Koco & Capponi, 2013), and the min-max measure in detection theory (Vincent, 1994); see Table 1.

We are interested in designing algorithms that are provably *consistent* for a given performance measure  $\psi$ , in that they converge (in probability) to the optimal  $\psi$ -performance as the training sample size increases:

**Definition 3** (Optimal  $\psi$ -performance). *For any function  $\psi : [0, 1]^{n \times n} \rightarrow \mathbb{R}_+$ , define the optimal  $\psi$ -performance w.r.t.  $D$  as the maximal  $\psi$ -performance over all randomized classifiers:*

$$\mathcal{P}_D^{\psi,*} = \sup_{h: \mathcal{X} \rightarrow \Delta_n} \mathcal{P}_D^\psi[h].$$

<sup>2</sup>Another popular variant of the micro  $F_1$  involves averaging the entries of the ‘one-versus-all’ binary confusion matrices for all classes, and computing the  $F_1$  for the averaged matrix; as pointed out by Manning et al. (2008), this form of micro  $F_1$  effectively reduces to the 0-1 classification accuracy. Recently, Parambath et al. (2014) also considered a form of micro  $F_1$  similar to that used in the BioNLP challenge (the expression they use is slightly simpler than ours and differs slightly from the BioNLP performance measure; see Appendix A.1 in the supplementary material).

**Definition 4** ( $\psi$ -regret). *For any classifier  $h$  and function  $\psi : [0, 1]^{n \times n} \rightarrow \mathbb{R}_+$ , define the  $\psi$ -regret of  $h$  w.r.t.  $D$  as the difference between its  $\psi$ -performance and the optimal:*

$$\mathcal{P}_D^{\psi,*} - \mathcal{P}_D^\psi[h].$$

**Definition 5** ( $\psi$ -consistent algorithm). *For any function  $\psi : [0, 1]^{n \times n} \rightarrow \mathbb{R}_+$ , say a multiclass algorithm  $\mathcal{A}$  that given a training sample  $S$  returns a classifier  $\mathcal{A}(S) : \mathcal{X} \rightarrow \Delta_n$  is  $\psi$ -consistent w.r.t.  $D$  if  $\forall \epsilon > 0$ :*

$$\mathbf{P}_{S \sim D^m}(\mathcal{P}_D^{\psi,*} - \mathcal{P}_D^\psi[\mathcal{A}(S)] > \epsilon) \rightarrow 0 \text{ as } m \rightarrow \infty.$$

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

$$\hat{\mathbf{C}}_{ij}^S[h] = \frac{1}{m} \sum_{k=1}^m \mathbf{1}(y_k = i, h(x_k) = j).$$

As a first step towards designing  $\psi$ -consistent algorithms, we start by examining the form of  $\psi$ -optimal classifiers.

### 3. Bayes Optimal Classifiers

For loss-based performance measures, it is well known that any classifier that always picks a class that minimizes the expected loss conditioned on the instance is optimal:

**Proposition 6.** *Let  $\mathbf{L} \in [0, 1]^{n \times n}$  be a loss matrix and  $\psi^{\mathbf{L}} : [0, 1]^{n \times n} \rightarrow \mathbb{R}_+$  be the corresponding loss-based performance measure,  $\psi^{\mathbf{L}}(\mathbf{C}) = 1 - \langle \mathbf{L}, \mathbf{C} \rangle$  (see Example 1). Then any (deterministic) classifier  $h^*$  satisfying*

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

*is a  $\psi^{\mathbf{L}}$ -optimal classifier, i.e.  $\mathcal{P}_D^{\mathbf{L}}[h^*] = \mathcal{P}_D^{\mathbf{L},*}$ .*

For binary performance measures expressed as functions of TN, FP, FN and TP (see Example 2), the following two results on the form of Bayes optimal classifiers for ‘ratio-of-linear’ binary performance measures and ‘monotonic’ binary performance measures, respectively, are known:

**Theorem 7** ((Koyejo et al., 2014)). *Let  $\mathcal{Y} = \{-1, 1\}$  and let  $\psi : [0, 1]^{2 \times 2} \rightarrow \mathbb{R}_+$  be a ratio-of-linear performance measure of the form  $\psi(\mathbf{C}) = \frac{a_{11}\text{TP} + a_{10}\text{FP} + a_{01}\text{FN} + a_{00}\text{TN}}{b_{11}\text{TP} + b_{10}\text{FP} + b_{01}\text{FN} + b_{00}\text{TN}}$  for some  $a_{ij}, b_{ij} \in \mathbb{R}$ . Then  $\exists$  a  $\psi$ -optimal classifier of one of the following forms:  $h^*(x) = \operatorname{sign}(\eta_1(x) - \theta_D^*)$  or  $h^*(x) = \operatorname{sign}(\theta_D^* - \eta_1(x))$ , where  $\theta_D^* \in [0, 1]$  depends on  $a_{ij}$ ’s and  $b_{ij}$ ’s, and on the optimal  $\psi$ -performance  $\mathcal{P}_D^{\psi,*}$ .<sup>3,4</sup>*

<sup>3</sup>The ratio-of-linear performance measures considered by Koyejo et al. (2014) have additional constant terms in the numerator and denominator; since the entries of a confusion matrix sum up to 1, these terms can be absorbed in the coefficients  $a_{ij}$ ’s,  $b_{ij}$ ’s.

<sup>4</sup>The original result of Koyejo et al. (2014) makes a continuity assumption on the marginal distribution  $\mu$ ; as we shall see in Theorem 11, the result holds even without this assumption.

**Theorem 8** ((Narasimhan et al., 2014)). Let  $\mathcal{Y} = \{-1, 1\}$  and let  $\psi : [0, 1]^{2 \times 2} \rightarrow \mathbb{R}_+$  be a continuous performance measure that is monotonically increasing in TP and TN and non-increasing in FP and FN. Let  $D$  be such that the CDF of the random variable  $\eta_1(X)$ ,  $\mathbf{P}(\eta_1(X) \leq z)$ , is continuous for all  $z \in (0, 1)$ . Then  $\exists$  a  $\psi$ -optimal classifier of the form  $h^*(x) = \text{sign}(\eta_1(x) - \theta_D^*)$  for some  $\theta_D^* \in [0, 1]$ .

In order to understand optimal classifiers for more general multiclass performance measures  $\psi$ , we will find it useful to view the optimal  $\psi$ -performance as the maximal value over all feasible *confusion matrices*:

**Definition 9** (Feasible confusion matrices). Define the set of feasible confusion matrices w.r.t.  $D$  as the set of all confusion matrices achieved by some randomized classifier:

$$\mathcal{C}_D = \{\mathbf{C}^D[h] : h : \mathcal{X} \rightarrow \Delta_n\}.$$

**Proposition 10.**  $\mathcal{C}_D$  is a convex set.

The set  $\mathcal{C}_D$  will play an important role in both our analysis of optimal classifiers and the subsequent development of consistent algorithms. Clearly, we can write

$$\mathcal{P}_D^{\psi,*} = \sup_{\mathbf{C} \in \mathcal{C}_D} \psi(\mathbf{C}). \quad (1)$$

While it is not clear if a classifier achieving this Bayes optimal performance exists in general, we show below that for ‘ratio-of-linear’ performance measures  $\psi$ , and for ‘monotonic’ performance measures  $\psi$  under a mild continuity assumption on  $D$ , an optimal classifier does indeed exist, and moreover, in each case, a  $\psi$ -optimal classifier can be obtained by finding a certain loss-based optimal classifier.

**Theorem 11** (Form of Bayes optimal classifier for ratio-of-linear  $\psi$ ). Let  $\psi : [0, 1]^{n \times n} \rightarrow \mathbb{R}_+$  be a ratio-of-linear performance measure of the form  $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$  for some  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$  with  $\langle \mathbf{B}, \mathbf{C} \rangle > 0 \forall \mathbf{C} \in \mathcal{C}_D$ . Let  $t_D^* = \mathcal{P}_D^{\psi,*}$ . Let  $\tilde{\mathbf{L}}^* = -(\mathbf{A} - t_D^* \mathbf{B})$ , and let  $\mathbf{L}^* \in [0, 1]^{n \times n}$  be obtained by scaling and shifting  $\tilde{\mathbf{L}}^*$  so its entries lie in  $[0, 1]$ . Then any classifier that is  $\psi^{\mathbf{L}^*}$ -optimal is also  $\psi$ -optimal.

**Lemma 12** (Existence of Bayes optimal classifier for monotonic  $\psi$ ). Let  $D$  be such that the probability measure associated with the random vector  $\boldsymbol{\eta}(X) = (\eta_1(X), \dots, \eta_n(X))^\top$  is absolutely continuous w.r.t. the base probability measure associated with the uniform distribution over  $\Delta_n$ , and let  $\psi$  be a performance measure that is differentiable and bounded over  $\mathcal{C}_D$ , and is monotonically increasing in  $C_{ii}$  for each  $i$  and non-increasing in  $C_{ij}$  for all  $i \neq j$ . Then  $\exists h^* : \mathcal{X} \rightarrow \Delta_n$  s.t.  $\mathcal{P}_D^{\psi}[h^*] = \mathcal{P}_D^{\psi,*}$ .

**Theorem 13** (Form of Bayes optimal classifier for monotonic  $\psi$ ). Let  $D, \psi$  satisfy the conditions of Lemma 12. Let  $h^* : \mathcal{X} \rightarrow \Delta_n$  be a  $\psi$ -optimal classifier and let  $\mathbf{C}^* = \mathbf{C}^D[h^*]$ . Let  $\tilde{\mathbf{L}}^* = -\nabla \psi(\mathbf{C}^*)$ , and let  $\mathbf{L}^* \in [0, 1]^{n \times n}$  be obtained by scaling and shifting  $\tilde{\mathbf{L}}^*$  so its entries lie in  $[0, 1]$ . Then any classifier that is  $\psi^{\mathbf{L}^*}$ -optimal is also  $\psi$ -optimal.

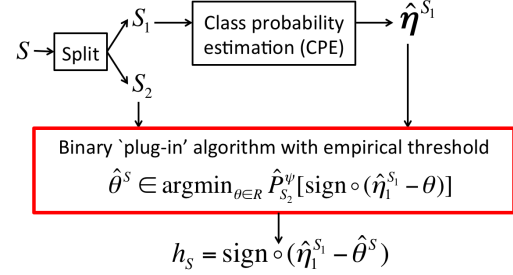


Figure 2. ‘Plug-in’ algorithm used for binary performance measures  $\psi$  (Koyejo et al., 2014; Narasimhan et al., 2014). In practice, one searches over  $O(|S_2|)$  values of the threshold  $\theta$ . In the multiclass case, such a method requires searching over an exponential number of loss matrices and is computationally intractable.

Theorems 11 and 13 generalize the results of Theorems 7 and 8 to the multiclass case; indeed, in the binary setting, classifiers that threshold the class probability function are known to be optimal for loss-based performance measures (Elkan, 2001). Moreover, by virtue of Proposition 6, Theorems 11 and 13 also imply that under the above conditions, one can always find a *deterministic* classifier that achieves the  $\psi$ -optimal performance.<sup>5</sup> Note that all performance measures in Table 1 are ‘monotonic’ as in Theorem 13; the micro  $F_1$  also has a ‘ratio-of-linear’ form as in Theorem 11.

The above results do not directly yield an algorithm since the linear performance measures  $\psi^{\mathbf{L}^*}$  that they suggest require knowledge of the optimal performance value  $\mathcal{P}_D^{\psi,*}$  in the ratio-of-linear case, or a  $\psi$ -optimal classifier  $h^*$  or  $\psi$ -optimal confusion matrix  $\mathbf{C}^*$  in the monotonic case. Nevertheless, a naïve algorithmic approach suggested by the above results is to search over a large range of  $n \times n$  loss matrices  $\mathbf{L}$ , estimate a  $\psi^{\mathbf{L}}$ -optimal classifier for each such  $\mathbf{L}$ , and select among these a classifier that yields maximal  $\psi$ -performance (e.g. on a held-out validation data set). This is the analogue of ‘plug-in’ type methods for binary performance measures, where one searches over possible thresholds on the (estimated) class probability function (see Figure 2). However, while the binary case involves a search over values for a single threshold parameter, in the multiclass case, searching over a suitable range of  $n \times n$  loss matrices  $\mathbf{L}$  in general requires time exponential in  $n^2$ , and for large  $n$  is computationally intractable.<sup>6</sup>

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

<sup>5</sup>This is not true in general; e.g. see Example 5 in Appendix B.1 for a setting where one needs a randomized classifier to achieve the optimal performance.

<sup>6</sup>For the special case of ratio-of-linear performance measures  $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$ , one can restrict the search to loss matrices of the form  $\mathbf{L} = -(\mathbf{A} - t\mathbf{B})$  for  $t \in \mathbb{R}$ , and can search over the single parameter  $t$ ; indeed, this is precisely what Parambath et al. (2014) do in the context of optimizing  $F$ -measures.



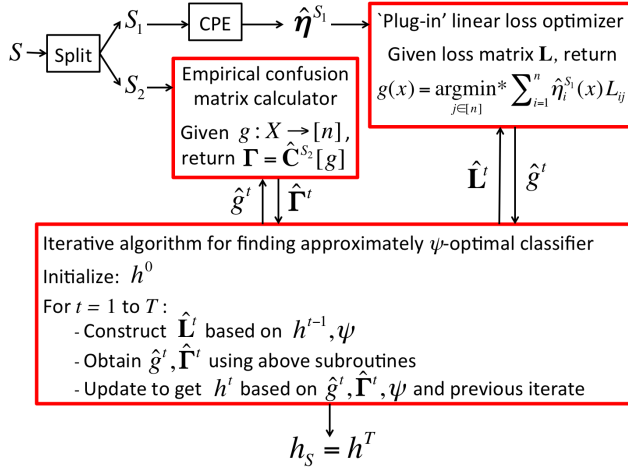


Figure 3. Overall framework of the multiclass learning algorithms proposed in this paper. The algorithms solve (approximately)  $\max_{\mathbf{C} \in \mathcal{C}_D} \psi(\mathbf{C})$ , by using an optimization method that on each iteration requires only solving a linear loss minimization problem and calculating an empirical confusion matrix, both of which can be done efficiently. Details of the constructions and updates depend on the underlying optimization method.

## 4. Algorithms

We design algorithms to search for  $\psi$ -optimal classifiers via a search over the set of feasible confusion matrices  $\mathcal{C}_D$ . While  $\mathcal{C}_D$  is a convex set, it is not available directly to the learner: not only is  $D$  unknown, but more fundamentally, the set of all confusion matrices is hard to characterize. On the other hand, given the class probability function  $\eta : \mathcal{X} \rightarrow \Delta_n$  – or an estimate  $\hat{\eta} : \mathcal{X} \rightarrow \Delta_n$  – one operation that is easy to perform is to find an optimal classifier for a linear loss  $\mathbf{L}$ : one simply returns the classifier  $g : \mathcal{X} \rightarrow [n]$  given by  $g(x) = \arg\min_{j \in [n]} \sum_{i=1}^n \hat{\eta}_i(x) L_{ij}$ . Moreover, given a classifier  $g$  and the distribution  $D$  – or a finite sample  $S$  – it is easy to calculate the confusion matrix of  $g$ : one simply computes for each  $i, j$  the proportion of examples  $(x, y)$  for which  $y = i$  and  $g(x) = j$ . In the following, we will design learning algorithms based on iterative optimization methods that do not require access to the full constraint set, but rather seek to (approximately, based on the training sample  $S$ ) solve the optimization problem  $\max_{\mathbf{C} \in \mathcal{C}_D} \psi(\mathbf{C})$  by making use of only the above two operations that can be performed efficiently.

In particular, we design two algorithms based on the above approach. The first applies to concave performance measures  $\psi$ , and makes use of the classical Frank-Wolfe optimization method, which solves general constrained convex optimization problems using only a linear minimization subroutine (Frank & Wolfe, 1956). The second algorithm applies to performance measures  $\psi$  that can be expressed as a ratio of linear functions,  $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$ ; in such cases, one can test whether the optimal value of  $\psi(\mathbf{C})$  exceeds a target value  $\gamma$  by again appealing to a linear min-

### Algorithm 1 Algorithm Based on Frank-Wolfe Method

- 1: **Input:**  $\psi : [0, 1]^{n \times n} \rightarrow \mathbb{R}_+$   
 $S = ((x_1, y_1), \dots, (x_m, y_m)) \in (\mathcal{X} \times [n])^m$
- 2: **Parameter:**  $\kappa \in \mathbb{N}$
- 3: Split  $S$  into  $S_1$  and  $S_2$  with sizes  $\lceil \frac{m}{2} \rceil$  and  $\lfloor \frac{m}{2} \rfloor$
- 4:  $\hat{\eta} = \text{CPE}(S_1)$
- 5: **Initialize:**  $h^0 : \mathcal{X} \rightarrow \Delta_n$ ,  $\hat{\mathbf{C}}^0 = \hat{\mathbf{C}}^{S_2}[h^0]$
- 6: **For**  $t = 1$  **to**  $T = \kappa m$  **do**
- 7:  $\hat{\mathbf{L}}^t = -\nabla \psi(\hat{\mathbf{C}}^{t-1})$ , scaled and shifted to  $[0, 1]^{n \times n}$
- 8: Obtain  $\hat{g}^t : x \mapsto \arg\min_{j \in [n]} \sum_{i=1}^n \hat{\eta}_i(x) \hat{L}_{ij}^t$
- 9:  $\hat{\mathbf{\Gamma}}^t = \hat{\mathbf{C}}^{S_2}[\hat{g}^t]$
- 10:  $h^t = (1 - \frac{2}{t+1})h^{t-1} + \frac{2}{t+1}\hat{g}^t$
- 11:  $\hat{\mathbf{C}}^t = (1 - \frac{2}{t+1})\hat{\mathbf{C}}^{t-1} + \frac{2}{t+1}\hat{\mathbf{\Gamma}}^t$
- 12: **end For**
- 13: **Output:**  $h_S^{\text{FW}} = h^T : \mathcal{X} \rightarrow \Delta_n$

imization subroutine, leading to an efficient binary search type algorithm based on the bisection method.

Both algorithms divide the input training sample  $S$  into a part  $S_1$  used for obtaining a class probability estimate  $\hat{\eta}^{S_1}$ , and a part  $S_2$  used for calculating empirical confusion matrices. On each iteration  $t$ , the algorithms implicitly maintain a confusion matrix  $\mathbf{C}^t = \mathbf{C}^D[h^t] \in \mathcal{C}_D$  by maintaining a (possibly randomized) classifier  $h^t$ , construct a linear loss  $\mathbf{L}^t$  based on  $\psi$  and the underlying optimization method (either the Frank-Wolfe method or the bisection method), solve a linear minimization problem that finds a ‘plug-in’ optimal classifier for this loss w.r.t. the class probability estimate  $\hat{\eta}^{S_1}$ , calculate the empirical confusion matrix corresponding to this classifier using  $S_2$ , and then update; after  $T$  iterations, the final classifier  $h^T$  is returned. The overall framework is summarized in Figure 3.

#### 4.1. Algorithm Based on Frank-Wolfe Method

The first algorithm that we describe uses the classical Frank-Wolfe method for constrained convex optimization (Frank & Wolfe, 1956) to learn a (randomized) classifier for performance measures  $\psi$  that are concave over  $\mathcal{C}_D$ , such as the G-mean measure in Example 3 (and the H-mean and Q-mean in Table 1). An ideal version of the algorithm for exactly solving  $\max_{\mathbf{C} \in \mathcal{C}_D} \psi(\mathbf{C})$  would maintain iterates  $\mathbf{C}^t \in \mathcal{C}_D$ , compute  $\mathbf{L}^t = -\nabla \psi(\mathbf{C}^{t-1})$ , solve exactly the resulting linear minimization problems  $\min_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{L}^t, \mathbf{C} \rangle$ , and update  $\mathbf{C}^t$  accordingly. As shown in Algorithm 1, the learning algorithm we propose maintains  $\mathbf{C}^t \in \mathcal{C}_D$  implicitly via  $h^t$ , and performs approximate sample-based computations in solving the linear minimization problems and computing confusion matrices. The final (randomized) classifier output by the algorithm is a convex combination of the classifiers learned across all the iterations.

The above algorithm does *not* amount to maximizing  $\psi$  over an empirical constraint set, but instead maximizes  $\psi$

directly over  $\mathcal{C}_D$ , with the associated linear minimization and confusion matrix calculation steps replaced with approximate, sample-based ones; this will be evident when we discuss consistency of the algorithm in Section 5.

#### 4.2. Algorithm Based on Bisection Method

The second algorithm we describe uses the bisection method (Boyd & Vandenberghe, 2004) and is designed for ratio-of-linear performance measures that can be written in the form  $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$  for some  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ , such as the micro  $F_1$ -measure in Example 4. For such performance measures, it is easy to see that  $\max_{\mathbf{C} \in \mathcal{C}_D} \psi(\mathbf{C}) \geq \gamma \iff \max_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{A} - \gamma \mathbf{B}, \mathbf{C} \rangle \geq 0$ ; thus, to test whether the optimal value of  $\psi$  is greater than  $\gamma$ , one can simply solve the linear minimization problem  $\min_{\mathbf{C} \in \mathcal{C}_D} -\langle \mathbf{A} - \gamma \mathbf{B}, \mathbf{C} \rangle$  and test the value of  $\psi$  at the resulting minimizer. Based on this observation, one can employ the bisection method to conduct a binary search for the maximal value (and maximizer) of  $\psi(\mathbf{C})$  using only a linear minimization subroutine.

An exact version of the algorithm would maintain  $\mathbf{C}^t \in \mathcal{C}_D$  together with lower and upper bounds  $\alpha^t$  and  $\beta^t$  on the maximal value of  $\psi$ , determine whether this maximal value is greater than the midpoint  $\gamma^t$  of these bounds using the linear minimization subroutine, and then update  $\mathbf{C}^t$  and  $\alpha^t, \beta^t$  accordingly. Again, as shown in Algorithm 2, the learning algorithm we propose maintains  $\mathbf{C}^t \in \mathcal{C}_D$  implicitly via  $h^t$ , and performs approximate sample-based computations in solving the linear minimization problems and computing confusion matrices. Since for ratio-of-linear performance measures there is always a deterministic classifier achieving the optimal performance (see Theorem 11), here it suffices to maintain deterministic classifiers  $h^t$ .<sup>7</sup>

The above bisection algorithm for ratio-of-linear performance measures generalizes and improves the method of Parambath et al. (2014), who use a similar idea in the context of optimizing F-measures but use a brute-force line search to estimate the optimal F-measure value; the bisection based algorithm, which essentially uses binary search, requires exponentially fewer computations.

### 5. Consistency

We now show that the algorithms proposed above are  $\psi$ -consistent. Our proofs rely on convergence guarantees of the underlying optimization methods, together with Lemmas 14 and 15 below, which yield approximation guaran-

<sup>7</sup>While the bisection based algorithm can be viewed as searching over a one-dimensional class of loss matrices, this is a special case; the Frank-Wolfe based algorithm for concave performance measures does not admit such an interpretation. Moreover, viewing the bisection algorithm as approximately solving  $\max_{\mathbf{C} \in \mathcal{C}_D} \psi(\mathbf{C})$  allows us to obtain consistency results in the same unified framework as the Frank-Wolfe based algorithm.

#### Algorithm 2 Algorithm Based on Bisection Method

- 1: **Input:**  $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$  with  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$   
 $S = ((x_1, y_1), \dots, (x_m, y_m)) \in (\mathcal{X} \times [n])^m$
- 2: **Parameter:**  $\kappa \in \mathbb{N}$
- 3: Split  $S$  into  $S_1$  and  $S_2$  with sizes  $\lceil \frac{m}{2} \rceil$  and  $\lfloor \frac{m}{2} \rfloor$
- 4:  $\hat{\eta} = \text{CPE}(S_1)$
- 5: **Initialize:**  $h^0 : \mathcal{X} \rightarrow [n], \alpha^0 = 0, \beta^0 = 1$
- 6: **For**  $t = 1$  to  $T = \kappa m$  **do**
- 7:    $\gamma^t = (\alpha^{t-1} + \beta^{t-1})/2$
- 8:    $\hat{\mathbf{L}}^t = -(\mathbf{A} - \gamma^t \mathbf{B})$ , scaled and shifted to  $[0, 1]^{n \times n}$
- 9:   Obtain  $\hat{g}^t : x \mapsto \arg\min_{j \in [n]} \sum_{i=1}^n \hat{\eta}_i(x) \hat{L}_{ij}^t$
- 10:    $\hat{\Gamma}^t = \hat{\mathbf{C}}^{S_2}[\hat{g}^t]$
- 11:   **If**  $\psi(\hat{\Gamma}^t) \geq \gamma^t$  **then**  $\alpha^t = \gamma^t, \beta^t = \beta^{t-1}, h^t = \hat{g}^t$
- 12:   **else**  $\alpha^t = \alpha^{t-1}, \beta^t = \gamma^t, h^t = h^{t-1}$
- 13: **end For**
- 14: **Output:**  $h_S^{\text{BS}} = h^T : \mathcal{X} \rightarrow [n]$

tees for the plug-in linear loss minimization and empirical confusion matrix calculation steps, respectively.

**Lemma 14** (L-regret of multiclass plug-in classifiers). *Let  $\hat{\eta} : \mathcal{X} \rightarrow \Delta_n$  and let  $\mathbf{L} \in [0, 1]^{n \times n}$ . Let  $\hat{h} : \mathcal{X} \rightarrow [n]$  be defined as  $\hat{h}(x) = \arg\min_{j \in [n]} \sum_{i=1}^n \hat{\eta}_i(x) L_{ij}$ . Then*

$$\mathcal{P}_D^{\mathbf{L},*} - \mathcal{P}_D^{\mathbf{L}}[\hat{h}] \leq \mathbf{E}_X [\|\hat{\eta}(X) - \eta(X)\|_1].$$

**Lemma 15** (Uniform convergence of confusion matrices). *Let  $\mathbf{q} : \mathcal{X} \rightarrow \Delta_n$  and let  $\mathcal{H}_{\mathbf{q}}$  be the set of (deterministic) classifiers  $h : \mathcal{X} \rightarrow [n]$  that satisfy  $h(x) = \arg\min_{j \in [n]} \sum_{i=1}^n q_i(x) L_{ij}$  for some  $\mathbf{L} \in [0, 1]^{n \times n}$ . Let  $S \in (\mathcal{X} \times [n])^m$  be drawn randomly from  $D^m$ . Let  $\delta \in (0, 1]$ . Then with probability  $\geq 1 - \delta$  (over  $S \sim D^m$ ),*

$$\sup_{h \in \mathcal{H}_{\mathbf{q}}} \|\mathbf{C}^D[h] - \hat{\mathbf{C}}^S[h]\|_{\infty} \leq C \sqrt{\frac{n^2 \log(n) \log(m) + \log(n^2/\delta)}{m}},$$

where  $C > 0$  is a distribution-independent constant.

#### 5.1. Consistency of Frank-Wolfe Based Algorithm

The following result bounds the  $\psi$ -regret of Algorithm 1 for any concave and smooth performance measure  $\psi$ :

**Theorem 16** ( $\psi$ -regret of Frank-Wolfe based algorithm). *Let  $\psi : [0, 1]^{n \times n} \rightarrow \mathbb{R}_+$  be concave over  $\mathcal{C}_D$ , and  $L$ -Lipschitz and  $\beta$ -smooth w.r.t. the  $\ell_1$  norm. Let  $S \in (\mathcal{X} \times [n])^m$  be drawn randomly from  $D^m$ . Let  $\hat{\eta} : \mathcal{X} \rightarrow \Delta_n$  be the CPE model learned in Algorithm 1 and  $h_S^{\text{FW}} : \mathcal{X} \rightarrow \Delta_n$  the classifier returned after  $\kappa m$  iterations. Let  $\delta \in (0, 1]$ . Then with probability  $\geq 1 - \delta$  (over  $S \sim D^m$ ),*

$$\mathcal{P}_D^{\psi,*} - \mathcal{P}_D^{\psi}[h_S^{\text{FW}}] \leq 4L \mathbf{E}_X [\|\hat{\eta}(X) - \eta(X)\|_1] + 4\sqrt{2}\beta n^2 C \sqrt{\frac{n^2 \log(n) \log(m) + \log(n^2/\delta)}{m}} + \frac{8\beta}{\kappa m + 2},$$

where  $C > 0$  is a distribution-independent constant.

The proof of Theorem 16 exploits Lemmas 14 and 15, together with the standard convergence guarantee for the Frank-Wolfe method (Jaggi, 2013). In particular, if the CPE model  $\hat{\eta}$  is learned by a CPE algorithm that guarantees  $\mathbb{E}_X[\|\hat{\eta}(X) - \eta(X)\|_1] \rightarrow 0$  as  $m \rightarrow \infty$ , as is done by any algorithm that minimizes a strictly proper composite multiclass loss over a suitably large function class (Vermet et al., 2011), then the above result yields  $\psi$ -consistency of the Frank-Wolfe based algorithm. For concave non-smooth performance measures  $\psi$  such as the G-mean, H-mean and Q-mean, Algorithm 1 can be applied to a suitable smooth approximation to  $\psi$ ; similar consistency guarantees can be shown in this case as well (see Appendix C.5).

## 5.2. Consistency of Bisection Based Algorithm

The following result bounds the  $\psi$ -regret of Algorithm 2 for ratio-of-linear performance measures  $\psi$ :

**Theorem 17** ( $\psi$ -regret of bisection based algorithm). *Let  $\psi : [0, 1]^{n \times n} \rightarrow \mathbb{R}_+$  be such that  $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$ , where  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ ,  $\sup_{\mathbf{C} \in \mathcal{C}_D} \psi(\mathbf{C}) \leq 1$ , and  $\min_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{B}, \mathbf{C} \rangle \geq b$  for some  $b > 0$ . Let  $S \in (\mathcal{X} \times [n])^m$  be drawn randomly from  $D^m$ . Let  $\hat{\eta} : \mathcal{X} \rightarrow \Delta_n$  be the CPE model learned in Algorithm 2 and  $h_S^{\text{BS}} : \mathcal{X} \rightarrow [n]$  the classifier returned after  $\kappa m$  iterations. Let  $\delta \in (0, 1]$ . Then with probability  $\geq 1 - \delta$  (over  $S \sim D^m$ ),*

$$\mathcal{P}_D^{\psi,*} - \mathcal{P}_D^{\psi}[h_S^{\text{BS}}] \leq 2\tau \mathbb{E}_X[\|\hat{\eta}(X) - \eta(X)\|_1] + 2\sqrt{2}C\tau \sqrt{\frac{n^2 \log(n) \log(m) + \log(n^2/\delta)}{m}} + 2^{-\kappa m},$$

where  $\tau = \frac{1}{b}(\|\mathbf{A}\|_1 + \|\mathbf{B}\|_1)$  and  $C > 0$  is a distribution-independent constant.

In this case, the proof of Theorem 17 exploits Lemmas 14 and 15, together with the well-known convergence guarantee for the bisection method (Boyd & Vandenberghe, 2004). Again, if the CPE algorithm used guarantees  $\mathbb{E}_X[\|\hat{\eta}(X) - \eta(X)\|_1] \rightarrow 0$  as  $m \rightarrow \infty$ , then the above result yields  $\psi$ -consistency of the bisection based algorithm. As a concrete example, with such a CPE method, we have that Algorithm 2 is consistent for the micro  $F_1$ -measure.

## 6. Experiments

We evaluated the proposed Frank-Wolfe and bisection based algorithms on a variety of multiclass learning tasks that differed in terms of performance measure, type of data set, number of classes, etc. In experiments with the Frank-Wolfe based algorithm, we considered the G-mean, H-mean and Q-mean performance measures, all of which are concave; in experiments with the bisection based algorithm, we considered the micro  $F_1$ -measure, which has a ratio-of-linear form. In all cases, we compared these algorithms against the state-of-the-art SVM<sup>perf</sup> algorithm

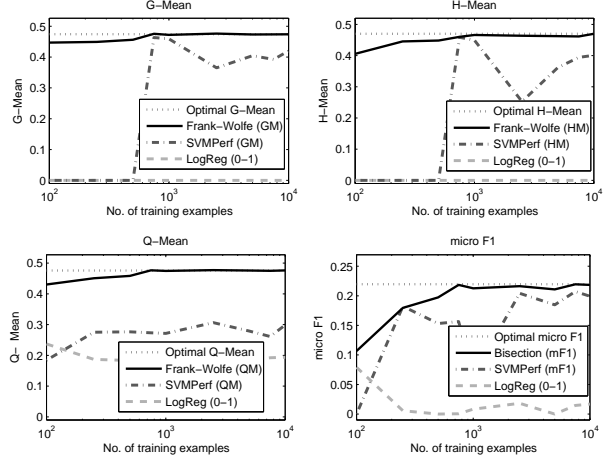


Figure 4. Convergence to Bayes optimal performance for G-mean, H-mean, Q-mean and micro  $F_1$  measures on synthetic data.

(Joachims, 2005) and a standard multiclass logistic regression algorithm that optimizes 0-1 accuracy.<sup>8</sup> We note that the worst-case running time of SVM<sup>perf</sup> is exponential in the number of classes, and hence this method could not be scaled to data sets with large numbers of classes.

## 6.1. Convergence to Bayes Optimal Performance

In a first set of experiments, we tested the consistency behavior of the algorithms on a synthetic data set for which the Bayes optimal performance could be calculated. Specifically, we used a 3-class synthetic data set with instances in  $\mathcal{X} = \mathbb{R}^2$  generated as follows: examples were chosen from class 1 with probability 0.85, from class 2 with probability 0.1 and from class 3 with probability 0.05; instances in the three classes were then drawn from multivariate Gaussian distributions with means  $(1, 1)^\top$ ,  $(0, 0)^\top$ , and  $(-1, -1)^\top$ , respectively and with the same covariance matrix  $\begin{bmatrix} 5 & 1 \\ 1 & 5 \end{bmatrix}$ . The class probability function  $\eta : \mathbb{R}^2 \rightarrow \Delta_3$  for this distribution is a softmax of linear functions that can be computed in closed form (see Appendix D.1). Note that the distribution and all four performance measures considered satisfy the conditions of Theorem 13.

Figure 4 shows the performance of the different algorithms for the G-mean, H-mean, Q-mean and micro  $F_1$  measures. In all cases, we learned a linear classification model (see Appendix D.2 for an explanation). As can be seen, for the G-mean, H-mean and Q-mean measures, the Frank-Wolfe based algorithm converges to the Bayes optimal performance, while the other algorithms fail to be consistent. For the micro  $F_1$ -measure (for which class 1 was taken as the default class), the bisection algorithm converges to the Bayes optimal performance; SVM<sup>perf</sup> also seems to approach the optimal performance, but at a slower rate.

<sup>8</sup>The CPE method used in the Frank-Wolfe and bisection based algorithms was also based on multiclass logistic regression.

Table 2. Data sets used in experiments in Sections 6.2–6.4.

	Data set	# instances	# features	# classes
UCI	car	1728	21	4
	pageblocks	5473	10	5
	glass	214	9	6
	abalone	4177	10	12
IR	cora	2708	1433	4
	news20	12199	61188	4
	rcv1	15564	47236	11

 Table 3. Performance of Frank-Wolfe based algorithm for G-mean, H-mean and Q-mean measures on various UCI data sets. The symbol  $\times$  indicates the method did not complete after 96 hrs.

		car	pgblks	glass	abalone
G-mean	Frank-Wolfe	<b>0.945</b>	<b>0.908</b>	<b>0.680</b>	<b>0.223</b>
	SVM <sup>perf</sup>	0.792	0.796	0.431	$\times$
	LogReg (0-1)	0.911	0.691	0.146	0.000
H-mean	Frank-Wolfe	<b>0.945</b>	<b>0.904</b>	<b>0.632</b>	<b>0.197</b>
	SVM <sup>perf</sup>	0.880	0.574	0.381	$\times$
	LogReg (0-1)	0.909	0.631	0.143	0.000
Q-mean	Frank-Wolfe	<b>0.930</b>	<b>0.877</b>	<b>0.613</b>	<b>0.247</b>
	SVM <sup>perf</sup>	0.909	0.651	0.481	$\times$
	LogReg (0-1)	0.898	0.660	0.490	0.223

## 6.2. Performance of Frank-Wolfe on UCI Data Sets

Our next set of experiments evaluates the Frank-Wolfe based algorithm on a variety of real data sets taken from the UCI repository (Frank & Asuncion, 2010). The data sets varied in size and number of classes; in many cases, there was moderate to severe imbalance across the various classes, a setting in which the G-mean, H-mean and Q-mean performance measures are of interest. We show results here for four of the data sets (see Table 2); results on additional data sets can be found in Appendix D.2.

As before, we learned linear models with all the algorithms: (regularized) linear multiclass logistic regression as the CPE method in the Frank-Wolfe based algorithm, and linear SVM<sup>perf</sup> and linear 0-1 multiclass logistic regression as baselines. The results, averaged over 5 random 80%-20% train-test splits for each data set, are shown in Table 3 (in the case of the Abalone data set, which has 12 classes, the SVM<sup>perf</sup> method did not complete running after 96 hours). As can be seen, in practically all cases, the Frank-Wolfe based algorithm outperforms both baselines.

## 6.3. Performance of Bisection on IR Data Sets

Next, we evaluate the bisection based algorithm on three information retrieval (IR) data sets, where the micro  $F_1$ -measure is of interest: a version of the CoRA data set containing research papers categorized into 7 classes, the 20 Newsgroups data set containing newsgroup documents categorized into 20 classes, and the RCV1 data set containing news articles from Reuters categorized into 53 classes (Forman, 2003; Druck et al., 2008; Lewis et al., 2004). For each of these data sets, we considered learning tasks where

 Table 4. Performance of bisection based algorithm for micro  $F_1$ -measure on CoRA, 20 Newsgroups, and Reuters RCV1 data sets. The symbol  $\times$  indicates the method did not complete after 96 hrs.

		cora	news20	rcv1
Micro $F_1$	Bisection	<b>0.690</b>	<b>0.772</b>	<b>0.502</b>
	SVM <sup>perf</sup>	0.622	$\times$	$\times$
	LogReg (0-1)	0.687	0.770	0.428

 Table 5. Training times (in secs) for various algorithms on UCI and IR data sets. The symbol  $\times$  indicates the method did not complete after 96 hrs. See Appendix D.2 for more details.

		car	pgblks	glass	abalone
G-mean	Frank-Wolfe	1.96	5.89	0.27	7.31
	SVM <sup>perf</sup>	8327.5	63667.7	1302.8	$\times$
	LogReg (0-1)	0.59	1.70	0.07	3.84
		cora	news20	rcv1	
Micro $F_1$	Bisection	0.23	13.40	10.43	
	SVM <sup>perf</sup>	18095.98	$\times$	$\times$	
	LogReg (0-1)	0.08	19.04	11.88	

a subset of the original set of classes was viewed as ‘interesting’ for prediction purposes, and the remaining classes were merged into a single ‘default’ class (used as class 1 in evaluating the micro  $F_1$  measure); this led to 4 effective classes for the CoRA and 20 Newsgroups data sets, and 11 effective classes for the RCV1 data set (see Table 2).

Again, we learned linear models with all the algorithms. The results, averaged over 5 random 80%-20% train-test splits for each data set, are shown in Table 4 (here again, SVM<sup>perf</sup> failed to complete running after 96 hours on the 20 Newsgroups and RCV1 data sets). As can be seen, the bisection based algorithm consistently yields micro  $F_1$  values better than or comparable to the baseline methods.

## 6.4. Run-Time Comparisons

Finally, we compare the training times of the various algorithms. Table 5 shows the training times (in seconds) for the G-mean and micro  $F_1$  performance measures (see Appendix D.2 for training times for H-mean and Q-mean). As can be seen, both the Frank-Wolfe based algorithm and the bisection based algorithm proposed here are several orders of magnitude faster than SVM<sup>perf</sup>, particularly on data sets with large numbers of classes.

## 7. Conclusion

In practice, classifiers are often evaluated using complex performance measures given by arbitrary functions of the confusion matrix. This paper has developed a general framework for designing consistent multiclass algorithms for such settings, and has given two practical algorithms that apply to a wide range of complex multiclass performance measures used in practice. The algorithms outperform existing baselines; in addition, they are computationally efficient and scale well with the number of classes.



**Acknowledgements.** HN acknowledges support from a Google India PhD Fellowship. HGR acknowledges support from a TCS PhD Fellowship. SA acknowledges support from the Department of Science & Technology (DST) of the Indian Government under a Ramanujan Fellowship, from the Indo-US Science & Technology Forum (IUSSTF), and from Yahoo in the form of an unrestricted grant.

## References

- Bartlett, P.L., Jordan, M.I., and McAuliffe, J.D. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.
- Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36:929–965, 1989.
- Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004.
- Buffoni, D., Calauzènes, C., Gallinari, P., and Usunier, N. Learning scoring functions with order-preserving losses and standardized supervision. In *ICML*, 2011.
- Calauzènes, C., Usunier, N., and Gallinari, P. On the (non-)existence of convex, calibrated surrogate losses for ranking. In *NIPS*, 2012.
- Cossock, D. and Zhang, T. Statistical analysis of Bayes optimal subset ranking. *IEEE Transactions on Information Theory*, 54(11):5140–5154, 2008.
- Dembczynski, K., Waegeman, W., Cheng, W., and Hüllermeier, E. An exact algorithm for F-measure maximization. In *NIPS*, 2011.
- Dembczynski, K., Jachnik, A., Kotłowski, W., Waegeman, W., and Hüllermeier, E. Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In *ICML*, 2013.
- Druck, G., Mann, G., and McCallum, A. Learning from labeled features using generalized expectation criteria. In *SIGIR*, 2008.
- Duchi, J., Mackey, L., and Jordan, M. On the consistency of ranking algorithms. In *ICML*, 2010.
- Elkan, C. The foundations of cost-sensitive learning. In *IJCAI*, 2001.
- Forman, G. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.
- Frank, A. and Asuncion, A. UCI machine learning repository, 2010. URL <http://archive.ics.uci.edu/ml>.
- Frank, M. and Wolfe, P. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2): 95–110, 1956.
- Gao, W. and Zhou, Z.-H. On the consistency of multi-label learning. In *COLT*, 2011.
- Jaggi, M. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML*, 2013.
- Joachims, T. A support vector method for multivariate performance measures. In *ICML*, 2005.
- Kennedy, K., Namee, B.M., and Delany, S.J. Learning without default: A study of one-class classification and the low-default portfolio problem. In *ICAICS*, 2009.
- Kim, J.-D., Wang, Y., and Yasunori, Y. The genia event extraction shared task, 2013 edition - overview. *ACL*, 2013.
- Koco, S. and Capponi, C. On multi-class classification through the minimization of the confusion matrix norm. In *ACML*, 2013.
- Koyejo, O., Natarajan, N., Ravikumar, P., and Dhillon, I.S. Consistent binary classification with generalized performance metrics. In *NIPS*, 2014.
- Lawrence, S., Burns, I., Back, A., Tsoi, A.-C., and Giles, C.L. Neural network classification and prior class probabilities. In *Neural Networks: Tricks of the Trade*, LNCS, pp. 1524:299–313. 1998.
- Lee, Y., Lin, Y., and Wahba, G. Multicategory support vector machines: Theory and application to the classification of microarray data. *Journal of the American Statistical Association*, 99(465):67–81, 2004.
- Lewis, D.D. Evaluating text categorization. In *Proceedings of the Workshop on Speech and Natural Language*, HLT, 1991.
- Lewis, D.D., Yang, Y., Rose, T.G., and Li, F. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.
- Machart, P. and Ralaivola, L. Confusion matrix stability bounds for multiclass classification. Technical report, Aix-Marseille University, 2012.
- Manning, C. D., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- Menon, A.K., Narasimhan, H., Agarwal, S., and Chawla, S. On the statistical consistency of algorithms for binary classification under class imbalance. In *ICML*, 2013.

- Musicant, D.R., Kumar, V., and Ozgur, A. Optimizing F-measure with support vector machines. In *FLAIRS*, 2003.
- Narasimhan, H., Vaish, R., and Agarwal, S. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In *NIPS*, 2014.
- Parambath, S.A.P., Usunier, N., and Grandvalet, Y. Optimizing F-measures by cost-sensitive classification. In *NIPS*, 2014.
- Pires, B. Á., Szepesvari, C., and Ghavamzadeh, M. Cost-sensitive multiclass classification risk bounds. In *ICML*, 2013.
- Ralaivola, L. Confusion-based online learning and a passive-aggressive scheme. In *NIPS*, 2012.
- Ramaswamy, H. G. and Agarwal, S. Classification calibration dimension for general multiclass losses. In *NIPS*, 2012.
- Ramaswamy, H. G., Agarwal, S., and Tewari, A. Convex calibrated surrogates for low-rank loss matrices with applications to subset ranking losses. In *NIPS*, 2013.
- Ravikumar, P., Tewari, A., and Yang, E. On NDCG consistency of listwise ranking methods. In *AISTATS*, 2011.
- Steinwart, I. How to compare different loss functions and their risks. *Constructive Approximation*, 26:225–287, 2007.
- Sun, Y., Kamel, M.S., and Wang, Y. Boosting for learning multiple classes with imbalanced class distribution. In *ICDM*, 2006.
- Tewari, A. and Bartlett, P. L. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8:1007–1025, 2007.
- Vernet, E., Williamson, R. C., and Reid, M. D. Composite multiclass losses. In *NIPS*, 2011.
- Vincent, P.H. *An Introduction to Signal Detection and Estimation*. Springer-Verlag, 1994.
- Wang, S. and Yao, X. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(4):1119–1130, 2012.
- Xia, F., Liu, T.-Y., Wang, J., Zhang, W., and Li, H. Listwise approach to learning to rank: Theory and algorithm. In *ICML*, 2008.
- Ye, N., Chai, K.M.A., Lee, W.S., and Chieu, H.L. Optimizing F-measures: A tale of two approaches. In *ICML*, 2012.
- Zhang, T. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32(1):56–134, 2004a.
- Zhang, T. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research*, 5:1225–1251, 2004b.

# Consistent Multiclass Algorithms for Complex Performance Measures

## Supplementary Material

**Notations.** Let  $\lambda$  be the base measure over  $\Delta_n$  given by the uniform random variable (say  $U$ ) over  $\Delta_n$ . Hence, for all measurable  $\mathcal{A} \subseteq \Delta_n$ ,  $\lambda(\mathcal{A}) = \mathbf{P}(U \in \mathcal{A})$ . Also,  $\boldsymbol{\eta} : \mathcal{X} \rightarrow \Delta_n$  is the mapping that gives the conditional probability vector  $\boldsymbol{\eta}(x) = [\mathbf{P}(Y = 1 | X = x), \dots, \mathbf{P}(Y = n | X = x)]^\top \in \Delta_n$  for a given instance  $x \in \mathcal{X}$ . Let  $\nu$  be the probability measure over the simplex induced by the random variable  $\boldsymbol{\eta}(X)$ ; in particular, for all measurable  $\mathcal{A} \subseteq \Delta_n$ ,  $\nu(\mathcal{A}) = \mathbf{P}_{X \sim \mu}(\boldsymbol{\eta}(X) \in \mathcal{A})$ . For a matrix  $\mathbf{L} \in [0, 1]^{n \times n}$  we let  $\ell_1, \ell_2, \dots, \ell_n$  be the columns of  $\mathbf{L}$ . For any set  $\mathcal{A} \subseteq \mathbb{R}^d$ , the set  $\bar{\mathcal{A}}$  denotes its closure. For any vector  $\mathbf{v} \in \mathbb{R}^n$ , we let  $(\mathbf{v})_{(i)}$  denote the  $i^{\text{th}}$  element when the components of  $\mathbf{v}$  are sorted in ascending order. For any  $y \in [n]$ , we shall denote  $\text{rand}(y) = [\mathbf{1}(y = 1), \dots, \mathbf{1}(y = n)]^\top$ .

### A. Supplementary Material For Section 2 (Complex Performance Measures)

#### A.1. Details of Micro $F_1$ -measure

We consider the form of the micro  $F_1$  used in the BioNLP challenge (Kim et al., 2013), which treats class 1 as a ‘default’ class (in information extraction, this class pertains to examples for which no information is required to be extracted). One can then define the micro precision of a classifier  $h : \mathcal{X} \rightarrow [n]$  with confusion matrix  $\mathbf{C} = \mathbf{C}^D[h]$  as the probability of an instance being correctly labelled, given that it was assigned by  $h$  a class other than 1:

$$\text{microPrec}(\mathbf{C}) = \mathbf{P}(h(X) = Y | h(X) \neq 1) = \frac{\sum_{i=2}^n C_{ii}}{\sum_{i=2}^n \sum_{j=1}^n C_{ji}} = \frac{\sum_{i=2}^n C_{ii}}{1 - \sum_{i=1}^n C_{i1}}$$

Similarly, the micro recall of  $h$  can be defined as the probability of an instance being correctly labelled, given that its true class was not 1:

$$\text{microRec}(\mathbf{C}) = \mathbf{P}(h(X) = Y | Y \neq 1) = \frac{\sum_{i=2}^n C_{ii}}{\sum_{i=2}^n \sum_{j=1}^n C_{ij}} = \frac{\sum_{i=2}^n C_{ii}}{1 - \sum_{i=1}^n C_{1i}}.$$

The micro  $F_1$  that we analyze is the harmonic mean of the micro precision and micro recall given above:

$$\psi^{\text{micro}F_1}(\mathbf{C}) = \frac{2 \times \text{microPrec}(\mathbf{C}) \times \text{microRec}(\mathbf{C})}{\text{microPrec}(\mathbf{C}) + \text{microRec}(\mathbf{C})} = \frac{2 \sum_{i=2}^n C_{ii}}{2 - \sum_{i=1}^n C_{1i} - \sum_{i=1}^n C_{i1}}.$$

Note that the above performance measure can be written as a ratio-of-linear function:  $\psi^{\text{micro}F_1}(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$ , where  $A_{11} = 0, A_{ii} = 2, \forall i \neq 1, A_{ij} = 0, \forall i \neq j$ , and  $B_{11} = 0, B_{1i} = B_{i1} = 1, \forall i \neq 1, B_{ij} = 2, \forall i \neq j$ . Also, note that this performance measure satisfies the condition in Theorem 17 with  $\sup_{\mathbf{C} \in \mathcal{C}_D} \psi^{\text{micro}F_1}(\mathbf{C}) \leq 1$ , and  $\min_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{B}, \mathbf{C} \rangle \geq 1 - \pi_1 > 0$ , and hence Algorithm 2 is consistent for this performance measure.

Recently, Parambath et al. (2014) also considered a form of micro  $F_1$  similar to that used in the BioNLP challenge. The expression they use is slightly simpler than ours and differs slightly from the BioNLP performance measure:

$$\psi^{\text{micro}F_1}(\mathbf{C}) = \frac{2 \sum_{i=2}^n C_{ii}}{1 + \sum_{i=2}^n C_{ii} - C_{11}}.$$

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

### B. Supplementary Material for Section 3 (Bayes Optimal Classifiers)

#### B.1. Example Distribution Where the Optimal Classifier Needs to be Randomized

We present an example distribution where the optimal performance for the G-mean measure can be achieved only by a randomized classifier.

**Example 5** (Distribution where the optimal classifier needs to be randomized). Let  $D$  be a distribution over  $\{x\} \times \{1, 2\}$  with  $\eta_1(x) = \eta_2(x) = \frac{1}{2}$  and suppose we are interested in finding the optimal classifier for the  $G$ -mean performance measure (see Example 3) under  $D$ . The two deterministic classifiers for this setting, namely, one which predicts 1 on  $x$  and the other that predicts 2 on  $x$ , yield a  $G$ -mean of 0. However, the randomized classifier  $h^*(x) = [\frac{1}{2}, \frac{1}{2}]^\top$  has a  $G$ -mean value of  $\frac{1}{4} > 0$  and can be verified to be the unique optimal classifier for  $G$ -mean under  $D$ .

We next present the proofs for the theorems/lemmas/propositions in Section 3.

## B.2. Proof of Theorem 11

**Theorem** (Form of Bayes optimal classifier for ratio-of-linear  $\psi$ ). Let  $\psi : [0, 1]^{n \times n} \rightarrow \mathbb{R}_+$  be a ratio-of-linear performance measure of the form  $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$  for some  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$  with  $\langle \mathbf{B}, \mathbf{C} \rangle > 0 \ \forall \mathbf{C} \in \mathcal{C}_D$ . Let  $t_D^* = \mathcal{P}_D^{\psi, *}$ . Let  $\tilde{\mathbf{L}}^* = -(\mathbf{A} - t_D^* \mathbf{B})$ , and let  $\mathbf{L}^* \in [0, 1]^{n \times n}$  be obtained by scaling and shifting  $\tilde{\mathbf{L}}^*$  so its entries lie in  $[0, 1]$ . Then any classifier that is  $\psi^{\mathbf{L}^*}$ -optimal is also  $\psi$ -optimal.

In the following, we omit the subscript on  $t_D^*$  for easy of presentation. We first state the following lemma using which we prove the above theorem.

**Lemma 18.** Let  $\psi : [0, 1]^{n \times n} \rightarrow \mathbb{R}_+$  be such that  $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$ , for some matrices  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$  with  $\langle \mathbf{B}, \mathbf{C} \rangle > 0$  for all  $\mathbf{C} \in \mathcal{C}_D$ . Let  $t^* = \sup_{\mathbf{C} \in \mathcal{C}_D} \psi(\mathbf{C})$ . Then  $\sup_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{A} - t^* \mathbf{B}, \mathbf{C} \rangle = 0$ .

*Proof.* Define  $\phi : \mathbb{R} \rightarrow \mathbb{R}$  as  $\phi(t) = \sup_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{A} - t \mathbf{B}, \mathbf{C} \rangle$ . It is easy to see that  $\phi$  (being a point-wise supremum of linear functions) is convex, and hence a continuous function over  $\mathbb{R}$ . By definition of  $t^*$ , we have for all  $\mathbf{C} \in \mathcal{C}_D$ ,

$$\frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle} \leq t^* \quad \text{or equivalently} \quad \phi(t^*) = \langle \mathbf{A} - t^* \mathbf{B}, \mathbf{C} \rangle \leq 0.$$

Thus

$$\phi(t^*) = \sup_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{A} - t^* \mathbf{B}, \mathbf{C} \rangle \leq 0. \quad (2)$$

Also, for any  $t < t^*$ , there exists  $\mathbf{C} \in \mathcal{C}_D$  such that

$$\frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle} > t \quad \text{or equivalently} \quad \phi(t) = \langle \mathbf{A} - t \mathbf{B}, \mathbf{C} \rangle > 0.$$

Thus for all  $t < t^*$ ,

$$\phi(t) = \sup_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{A} - t \mathbf{B}, \mathbf{C} \rangle > 0.$$

Next, by continuity of  $\phi$ , for any monotonically increasing sequence of real numbers  $\{t_i\}_{i=1}^\infty$  converging to  $t^*$ , we have that  $\phi(t_i)$  converges to  $\phi(t^*)$ ; since for each  $t_i$  in this sequence  $\phi(t_i) > 0$ , at the  $t^*$ , we have that  $\phi(t^*) \geq 0$ . Along with Eq. (2), this gives us

$$\sup_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{A} - t^* \mathbf{B}, \mathbf{C} \rangle = \phi(t^*) = 0.$$

□

We next give the proof for Theorem 11

*Proof of Theorem 11.* Let  $h^* : \mathcal{X} \rightarrow \Delta_n$  be a  $\psi^{\mathbf{L}^*}$ -optimal classifier. We shall show that  $h^*$  is also  $\psi$ -optimal, which will also imply existence of the  $\psi$ -optimal classifier. Then we have

$$1 - \langle \mathbf{L}^*, \mathbf{C}^D[h^*] \rangle = \sup_{h: \mathcal{X} \rightarrow \Delta_n} 1 - \langle \mathbf{L}^*, \mathbf{C}^D[h] \rangle = \sup_{\mathbf{C} \in \mathcal{C}_D} 1 - \langle \mathbf{L}^*, \mathbf{C} \rangle.$$

Since  $\mathbf{L}^*$  is a scaled and translated version of  $\tilde{\mathbf{L}}^* = \mathbf{A} - t^* \mathbf{B}$  (where  $t^* = \mathcal{P}^{\psi, *}$ ), we further have

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



Now, from Lemma 18 we know that  $\langle \mathbf{A} - t^* \mathbf{B}, \mathbf{C}^D[h^*] \rangle = 0$ . Hence,

$$\frac{\langle \mathbf{A}, \mathbf{C}^D[h^*] \rangle}{\langle \mathbf{B}, \mathbf{C}^D[h^*] \rangle} = t^*,$$

or equivalently,

$$\psi(\mathbf{C}^D[h^*]) = \sup_{\mathbf{C} \in \mathcal{C}_D} \psi(\mathbf{C}).$$

Thus  $h^*$  is also  $\psi$ -optimal, which completes the proof.  $\square$

### B.3. Proof of Proposition 10

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

*Proof.* Let  $\mathbf{C}_1, \mathbf{C}_2 \in \mathcal{C}_D$ . Let  $\lambda \in [0, 1]$ . We will show that  $\lambda \mathbf{C}_1 + (1 - \lambda) \mathbf{C}_2 \in \mathcal{C}_D$ .

By definition of  $\mathcal{C}_D$ , there exists randomized classifiers  $h_1, h_2 : \mathcal{X} \rightarrow \Delta_n$  such that  $\mathbf{C}_1 = \mathbf{C}^D[h_1]$  and  $\mathbf{C}_2 = \mathbf{C}^D[h_2]$ .

Consider the randomized classifier  $h^\lambda : \mathcal{X} \rightarrow \Delta_n$  defined as

$$h^\lambda(x) = \lambda h_1(x) + (1 - \lambda) h_2(x).$$

It can be seen that

$$\mathbf{C}^D[h^\lambda] = \lambda \mathbf{C}_1 + (1 - \lambda) \mathbf{C}_2.$$

$\square$

### B.4. Supporting Technical Lemmas For Lemma 12 and Theorem 13

In this subsection we give some supporting technical lemmas which will be useful in the proofs for Lemma 12 and Theorem 13.

**Lemma 19 (Confusion matrix as an integration).** Let  $\mathbf{f} : \Delta_n \rightarrow \Delta_n$ . Then

$$\mathbf{C}^D[\mathbf{f} \circ \boldsymbol{\eta}] = \int_{\mathbf{p} \in \Delta_n} \mathbf{p}(\mathbf{f}(\mathbf{p}))^\top d\nu(\mathbf{p}).$$

*Proof.*

$$\begin{aligned} C_{i,j}^D[\mathbf{f} \circ \boldsymbol{\eta}] &= \mathbf{E}_{(X,Y) \sim D}[f_j(\boldsymbol{\eta}(X)) \cdot \mathbf{1}(Y = i)] \\ &= \mathbf{E}_{\mathbf{p} \sim \nu} \mathbf{E}_{(X,Y) \sim D}[f_j(\mathbf{p}) \cdot \mathbf{1}(Y = i) | \boldsymbol{\eta}(X) = \mathbf{p}] \\ &= \mathbf{E}_{\mathbf{p} \sim \nu}[p_i \cdot f_j(\mathbf{p})]. \end{aligned}$$

$\square$

**Proposition 20 (Sufficiency of conditional probability).** Let  $D$  be a distribution over  $\mathcal{X} \times \mathcal{Y}$ . For any randomized classifier  $h : \mathcal{X} \rightarrow \Delta_n$  there exists another randomized classifier  $h' : \mathcal{X} \rightarrow \Delta_n$  such that  $\mathbf{C}^D[h] = \mathbf{C}^D[h']$  and  $h'$  is such that  $h' = \mathbf{f} \circ \boldsymbol{\eta}$ , for some  $\mathbf{f} : \Delta_n \rightarrow \Delta_n$ .

*Proof.* Let  $h : \mathcal{X} \rightarrow \Delta_n$ . Define  $\mathbf{f} : \Delta_n \rightarrow \Delta_n$  as follows,

$$\mathbf{f}(\mathbf{p}) = \mathbf{E}_{X \sim \mu}[h(X) | \boldsymbol{\eta}(X) = \mathbf{p}].$$

We then have for any  $i, j \in [n]$  that,

$$\begin{aligned} C_{i,j}^D[h] &= \mathbf{E}_{(X,Y) \sim D}[h_j(X) \cdot \mathbf{1}(Y = i)] \\ &= \mathbf{E}_{\mathbf{p} \sim \nu} \mathbf{E}_{(X,Y) \sim D}[h_j(X) \cdot \mathbf{1}(Y = i) | \boldsymbol{\eta}(X) = \mathbf{p}] \end{aligned}$$

$$\begin{aligned}
 &= \mathbf{E}_{\mathbf{p} \sim \nu} [\mathbf{E}_{(X,Y) \sim D} [h_j(X) | \boldsymbol{\eta}(X) = \mathbf{p}] \cdot \mathbf{E}_{(X,Y) \sim D} [\mathbf{1}(Y = i) | \boldsymbol{\eta}(X) = \mathbf{p}]] \\
 &= \mathbf{E}_{\mathbf{p} \sim \nu} [f_j(\mathbf{p}) \cdot p_i] \\
 &= C_{i,j}^D[\mathbf{f} \circ \boldsymbol{\eta}]
 \end{aligned}$$

where the third equality follows because, given  $\boldsymbol{\eta}(X)$ , the random variables  $X$  and  $Y$  are independent, and the last inequality follows from Lemma 19.  $\square$

**Lemma 21 (Continuity of the  $C^D$  mapping).** *Let  $D$  be a distribution over  $\mathcal{X} \times \mathcal{Y}$ . Let  $\mathbf{f}_1, \mathbf{f}_2 : \Delta_n \rightarrow \Delta_n$ . Then*

$$\|C^D[\mathbf{f}_1 \circ \boldsymbol{\eta}] - C^D[\mathbf{f}_2 \circ \boldsymbol{\eta}]\|_1 \leq \int_{\mathbf{p} \in \Delta_n} \|\mathbf{f}_1(\mathbf{p}) - \mathbf{f}_2(\mathbf{p})\|_1 d\nu(\mathbf{p}).$$

*Proof.* Let  $\mathbf{f}_1, \mathbf{f}_2 : \Delta_n \rightarrow \Delta_n$

$$\begin{aligned}
 C^D[\mathbf{f}_1 \circ \boldsymbol{\eta}] - C^D[\mathbf{f}_2 \circ \boldsymbol{\eta}] &= \int_{\mathbf{p} \in \Delta_n} \mathbf{p}(\mathbf{f}_1(\mathbf{p}) - \mathbf{f}_2(\mathbf{p}))^\top d\nu(\mathbf{p}) \\
 \|C^D[\mathbf{f}_1 \circ \boldsymbol{\eta}] - C^D[\mathbf{f}_2 \circ \boldsymbol{\eta}]\|_1 &\leq \int_{\mathbf{p} \in \Delta_n} \|\mathbf{p}(\mathbf{f}_1(\mathbf{p}) - \mathbf{f}_2(\mathbf{p}))^\top\|_1 d\nu(\mathbf{p}) \\
 &= \int_{\mathbf{p} \in \Delta_n} \|\mathbf{p}\|_1 \|\mathbf{f}_1(\mathbf{p}) - \mathbf{f}_2(\mathbf{p})\|_1 d\nu(\mathbf{p}) \\
 &= \int_{\mathbf{p} \in \Delta_n} \|\mathbf{f}_1(\mathbf{p}) - \mathbf{f}_2(\mathbf{p})\|_1 d\nu(\mathbf{p}).
 \end{aligned}$$

$\square$

**Lemma 22 (Volume of a inverse linear map of an interval).** *Let  $d > 0$  be any integer. Let  $\mathcal{V} \subseteq \mathbb{R}^d$  be compact and convex. Let  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  be an affine function such that it is non-constant over  $\mathcal{V}$ . Let  $V$  be a vector valued random variable taking values uniformly over  $\mathcal{V}$ . Then, there exists a constant  $\alpha > 0$  such that for all  $c \in \mathbb{R}$  and  $\epsilon \in \mathbb{R}_+$  we have*

$$\mathbf{P}(f(V) \in [c, c + \epsilon]) \leq \alpha \epsilon.$$

*Proof.* Let us assume for now that affine hull of  $\mathcal{V}$  is the entire space  $\mathbb{R}^d$ .

For any integer  $i$  and set  $\mathcal{A}$ , let  $\text{vol}_i(\mathcal{A})$  denote the  $i$ -th dimensional volume of the set  $\mathcal{A}$ . Note that  $\text{vol}_i(\mathcal{A})$  is undefined if the affine-hull dimension of  $\mathcal{A}$  is greater than  $i$  and is equal to zero if the affine-hull dimension of  $\mathcal{A}$  is lesser than  $i$ .

For any  $r > 0$  and any integer  $i > 0$  let  $B_i(r) \subseteq \mathbb{R}^i$  denote the set  $B_i(r) = \{\mathbf{x} \in \mathbb{R}^i : \|\mathbf{x}\|_2 \leq r\}$ . Also let  $R$  be the smallest value such that  $\mathcal{V} \subseteq B_d(R)$ .

Let the affine function  $f$  be such that for all  $\mathbf{x} \in \mathbb{R}^d$ , the value  $f(\mathbf{x}) = \mathbf{g}^\top \mathbf{x} + u$ . By the assumption of non-constancy of  $f$  on  $\mathcal{V}$  we have that  $\mathbf{g} \neq 0$ .

We now have that

$$\begin{aligned}
 \mathbf{P}(f(V) \in [c, c + \epsilon]) &= \frac{\text{vol}_d(\{\mathbf{v} \in \mathcal{V} : c - u \leq \mathbf{g}^\top \mathbf{v} \leq c - u + \epsilon\})}{\text{vol}_d(\mathcal{V})} \\
 &\leq \frac{\text{vol}_d(\{\mathbf{v} \in B_d(R) : c - u \leq \mathbf{g}^\top \mathbf{v} \leq c - u + \epsilon\})}{\text{vol}_d(\mathcal{V})} \\
 &\leq \epsilon \cdot \frac{\text{vol}_{d-1}(B_{d-1}(R))}{\text{vol}_d(\mathcal{V}) \|\mathbf{g}\|_2}.
 \end{aligned}$$

The last inequality follows from the observation that  $d$ -volume of a strip of a  $d$  dimensional sphere of radius  $r$  is at most the  $d - 1$  volume of a  $d - 1$  dimensional sphere of radius  $r$  times the width of the strip, and the width of the strip under consideration here is simply  $\frac{\epsilon}{\|\mathbf{g}\|_2}$ .

Finally, if the affine hull of  $\mathcal{V}$  is not the entire space  $\mathbb{R}^d$ , one can simply consider the affine-hull of  $\mathcal{V}$  to be the entire (lesser dimensional) space and all the above arguments hold with some affine transformations and a smaller  $d$ .  $\square$

**Lemma 23 (Fraction of instances with the best and second best prediction being similar in performance is small).**

Let  $\mathbf{L} \in [0, 1]^{n \times n}$  be such that no two columns are identical. Let the distribution  $D$  over  $\mathcal{X} \times \mathcal{Y}$  be such that the measure over conditional probabilities  $\nu$ , is absolutely continuous w.r.t. the base measure  $\lambda$ . Let  $c \geq 0$ . Let  $\mathcal{A}_c \subseteq \Delta_n$  be the set

$$\mathcal{A}_c = \{\mathbf{p} \in \Delta_n : (\mathbf{p}^\top \mathbf{L})_{(2)} - (\mathbf{p}^\top \mathbf{L})_{(1)} \leq c\}$$

Let  $r : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  be the function defined as

$$r(c) = \nu(\mathcal{A}_c) .$$

Then

- (a)  $r$  is a monotonically increasing function.
- (b) There exists a  $C > 0$  such that  $r$  is a continuous function over  $[0, C]$ .
- (c)  $r(0) = 0$ .

*Proof.* Part (a):

The fact that  $r$  is a monotonically increasing function is immediately obvious from the observation that  $\mathcal{A}_a \subseteq \mathcal{A}_b$  for any  $a < b$ .

Part (b):

Let

$$C = \frac{1}{2} \min\{d \in \mathbb{R} : \ell_y - \ell_{y'} = de \text{ for some } y, y' \in [n], y \neq y'\} ,$$

where  $\mathbf{e}$  is the all ones vector. If there exists no  $y, y'$  such that  $\ell_y - \ell_{y'}$  is a scalar multiple of  $\mathbf{e}$ , then we simply set  $C = \infty$ . Note that by our assumption on unequal columns on  $\mathbf{L}$ , we always have  $C > 0$ .

For any  $c > 0$  and  $y, y' \in [n]$  with  $y \neq y'$ , define the set  $\mathcal{A}_c^{y, y'}$  as

$$\mathcal{A}_c^{y, y'} = \{\mathbf{p} \in \Delta_n : \mathbf{p}^\top \ell_y - \mathbf{p}^\top \ell_{y'} \leq c\} .$$

For any  $c, \epsilon > 0$ , it can be clearly seen that

$$\begin{aligned} \nu(\mathcal{A}_{c+\epsilon}) - \nu(\mathcal{A}_c) &= \nu(\mathcal{A}_{c+\epsilon} \setminus \mathcal{A}_c) , \\ \mathcal{A}_{c+\epsilon} \setminus \mathcal{A}_c &\subseteq \bigcup_{y, y' \in [n], y \neq y'} (\mathcal{A}_{c+\epsilon}^{y, y'} \setminus \mathcal{A}_c^{y, y'}) , \\ \nu(\mathcal{A}_{c+\epsilon} \setminus \mathcal{A}_c) &\leq \sum_{y, y' \in [n], y \neq y'} \nu(\mathcal{A}_{c+\epsilon}^{y, y'} \setminus \mathcal{A}_c^{y, y'}) . \end{aligned}$$

Hence, our proof for continuity of  $r$  would be complete, if we show that  $\nu(\mathcal{A}_{c+\epsilon}^{y, y'} \setminus \mathcal{A}_c^{y, y'})$  goes to zero as  $\epsilon$  goes to zero for all  $y \neq y'$  and  $c \in [0, C]$ .

Let  $c \in [0, C]$  and  $y, y' \in [n]$  with  $y \neq y'$

$$\mathcal{A}_{c+\epsilon}^{y, y'} \setminus \mathcal{A}_c^{y, y'} = \{\mathbf{p} \in \Delta_n : c < \mathbf{p}^\top (\ell_y - \ell_{y'}) \leq c + \epsilon\} .$$

If  $\ell_y - \ell_{y'} = de$  for some  $d$ , we have that  $\mathbf{p}^\top (\ell_y - \ell_{y'}) = d$  and  $d > C$  by definition of  $C$ . Hence for small enough  $\epsilon$  the set  $\mathcal{A}_{c+\epsilon}^{y, y'} \setminus \mathcal{A}_c^{y, y'}$  is empty.

If  $\ell_y - \ell_{y'}$  is not a scalar multiple of  $\mathbf{e}$ , then  $\mathbf{p}^\top (\ell_y - \ell_{y'})$  is a non-constant linear function of  $\mathbf{p}$  over  $\Delta_n$ . From Lemma 22,  $\lambda(\mathcal{A}_{c+\epsilon}^{y, y'} \setminus \mathcal{A}_c^{y, y'})$  goes to zero as  $\epsilon$  goes to zero. And by the absolute continuity of  $\nu$  w.r.t.  $\lambda$ , we have  $\nu(\mathcal{A}_{c+\epsilon}^{y, y'} \setminus \mathcal{A}_c^{y, y'})$  goes to zero as  $\epsilon$  goes to zero.

As the above arguments hold for any  $c \in [0, C]$  and  $y, y' \in [n]$  with  $y \neq y'$ , the proof of part (b) is complete.

Part (c):

We have,

$$\mathcal{A}_0 \subseteq \bigcup_{y, y' \in [n], y \neq y'} (\mathcal{A}_0^{y, y'} \cap \mathcal{A}_0^{y', y}).$$

To show  $r(0) = 0$ , we show  $\lambda(\mathcal{A}_0^{y, y'} \cap \mathcal{A}_0^{y', y}) = 0$  for all  $y \neq y'$ . Let  $y, y' \in [n]$  with  $y \neq y'$ , then

$$(\mathcal{A}_0^{y, y'} \cap \mathcal{A}_0^{y', y}) = \{\mathbf{p} \in \Delta_n : \mathbf{p}^\top (\ell_y - \ell_{y'}) = 0\}.$$

If  $\ell_y - \ell_{y'} = d\mathbf{e}$  for some  $d \neq 0$ , the above set is clearly empty. If  $\ell_y - \ell_{y'}$  is not a scalar multiple of  $\mathbf{e}$ , then  $\mathbf{p}^\top (\ell_y - \ell_{y'})$  is a non-constant linear function of  $\mathbf{p}$  over  $\Delta_n$ , and hence by Lemma 22, we have that  $\lambda(\mathcal{A}_0^{y, y'} \cap \mathcal{A}_0^{y', y}) = 0$ . By the absolute continuity of  $\nu$  w.r.t.  $\lambda$  we have that  $\nu(\mathcal{A}_0^{y, y'} \cap \mathcal{A}_0^{y', y}) = 0$ .

As the above arguments hold for any  $y, y' \in [n]$  with  $y \neq y'$ , the proof of part (c) is complete.  $\square$

**Lemma 24 (The uniqueness of  $\psi^{\mathbf{L}}$ -optimal classifier).** *Let the distribution  $D$  over  $\mathcal{X} \times \mathcal{Y}$  be such that the measure over conditional probabilities  $\nu$ , is absolutely continuous w.r.t. the base measure  $\lambda$ . Let  $\mathbf{L} \in \mathbb{R}^{n \times n}$  be such that no two columns are identical. Then, all  $\psi^{\mathbf{L}}$  optimal classifiers have the same confusion matrix. i.e. the minimizer over  $\mathcal{C}_D$  of  $\langle \mathbf{L}, \mathbf{C} \rangle$  is unique.*

*Proof.* If  $x \in \mathcal{X}$  is such that  $\arg\min_{y \in [n]} \boldsymbol{\eta}(x)^\top \ell_y$  is a singleton, then any  $\psi^{\mathbf{L}}$ -optimal classifier  $h^*$  is such that

$$h^*(x) = \arg\min_{y \in [n]} \boldsymbol{\eta}(x)^\top \ell_y$$

We just show that the set of instances in  $\mathcal{X}$ , such that  $\arg\min_{y \in [n]} \boldsymbol{\eta}(x)^\top \ell_y$  is not a singleton, has measure zero. For any  $\mathbf{v} \in \mathbb{R}^n$ , let  $(\mathbf{v})_{(i)}$  be the  $i^{\text{th}}$  element when the components of  $\mathbf{v}$  are arranged in ascending order.

$$\begin{aligned} \mu(\{x \in \mathcal{X} : |\arg\min_{y \in [n]} \boldsymbol{\eta}(x)^\top \ell_y| > 1\}) &= \mu(\{x \in \mathcal{X} : (\boldsymbol{\eta}(x)^\top \mathbf{L})_{(1)} = (\boldsymbol{\eta}(x)^\top \mathbf{L})_{(2)}\}) \\ &= \nu(\{\mathbf{p} \in \Delta_n : (\mathbf{p}^\top \mathbf{L})_{(1)} = (\mathbf{p}^\top \mathbf{L})_{(2)}\}) \end{aligned}$$

Thus, by Lemma 23 (part c), we have that set of instances in  $\mathcal{X}$  such that  $\arg\min_{y \in [n]} \boldsymbol{\eta}(x)^\top \ell_y$  is not a singleton, has measure zero. Thus any pair of  $\psi^{\mathbf{L}}$ -optimal classifiers are same  $\mu$  almost everywhere, and hence all the  $\psi^{\mathbf{L}}$ -optimal classifiers have the same confusion matrix.  $\square$

Next we give the master Lemma which uses every result in this section, and will actually be the only tool in the proofs of Lemma 12 and Theorem 13.

**Lemma 25 (Master Lemma).** *Let the distribution  $D$  over  $\mathcal{X} \times \mathcal{Y}$  be such that the measure over conditional probabilities  $\nu$ , be absolutely continuous w.r.t. the base measure  $\lambda$ . Let  $\mathbf{L} \in [0, 1]^{n \times n}$  be such that no two columns are identical. Then,*

$$\arg\min_{\mathbf{C} \in \overline{\mathcal{C}_D}} \langle \mathbf{L}, \mathbf{C} \rangle = \arg\min_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{L}, \mathbf{C} \rangle.$$

Moreover, the above set is a singleton.

*Proof.* The first part of the proof where one shows  $\arg\min_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{L}, \mathbf{C} \rangle$  is a singleton is exactly what is given by Lemma 24. Let

$$\mathbf{C}^* = \mathbf{C}^D[h^*] = \arg\min_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{L}, \mathbf{C} \rangle.$$

The classifier  $h^* : \mathcal{X} \rightarrow \Delta_n$  is such that  $\mathbf{C}^D[h^*] = \mathbf{C}^*$ , and is fixed for convenience as the following classifier,

$$h^*(x) = \text{rand}(\arg\min_{y \in [n]} \boldsymbol{\eta}(x)^\top \ell_y).$$



Also let  $\mathbf{f}^* : \Delta_n \rightarrow \Delta_n$  be such that,  $h^* = \mathbf{f}^* \circ \eta$ , i.e.

$$\mathbf{f}^*(\mathbf{p}) = \text{rand}(\arg\min_{y \in [n]} \mathbf{p}^\top \boldsymbol{\ell}_y) .$$

In the rest of the proof we simply show that  $\mathbf{C}^*$  is the unique minimizer of  $\langle \mathbf{L}, \mathbf{C} \rangle$  over  $\overline{\mathcal{C}_D}$  as well. To do so, we first assume that  $\mathbf{C}' \in \arg\min_{\mathbf{C} \in \overline{\mathcal{C}_D}} \langle \mathbf{L}, \mathbf{C} \rangle$  and  $\mathbf{C}' \neq \mathbf{C}^*$ . We then go on to show a contradiction, the brief details of which are given below:

1. As  $\mathbf{C}' \neq \mathbf{C}^*$ , we have  $\|\mathbf{C}^* - \mathbf{C}'\|_1 = \xi > 0$ .
2. As  $\mathbf{C}' \in \overline{\mathcal{C}_D}$ , there exists a sequence of classifiers  $h_1, h_2, \dots$ , such that their confusion matrices converge to  $\mathbf{C}'$ .
3. The confusion matrices of the classifiers  $h_1, h_2, \dots$ , are bounded away from  $\mathbf{C}^*$  as  $\xi > 0$ .
4. Due to the continuity of the  $\mathbf{C}^D$  mapping (Lemma 21), the classifiers  $h_1, h_2, \dots$  are also bounded away from  $h^*$  – i.e. they must predict differently from  $h^*$  on a significant fraction of the instances.
5. Due to Lemma 23, we have that for most instances the second best prediction (in terms of loss  $\mathbf{L}$ ) is significantly worse than the best prediction. The classifiers  $h_1, h_2, \dots$  all predict differently from  $h^*$  (which always predicts the best label for any given instance) for a large fraction of the instances, hence they must predict a significantly worse label for a large fraction of instances.
6. From the above reasoning, the classifiers  $h_1, h_2, \dots$ , all perform worse by a constant additive factor than  $h^*$ , on the  $\psi^{\mathbf{L}}$  performance measure. But, as the confusion matrices of these classifiers converge to  $\mathbf{C}'$ , the  $\psi^{\mathbf{L}}$  performance of these classifiers must approach the optimal. Thus providing a contradiction.

The full details of the above sketch is given below.

Let

$$\|\mathbf{C}' - \mathbf{C}^*\|_1 = \xi > 0 .$$

As  $\mathbf{C}' \in \overline{\mathcal{C}_D}$ , we have that for all  $\epsilon > 0$ , there exists  $\mathbf{C}_\epsilon \in \mathcal{C}_D$ , such that  $\|\mathbf{C}_\epsilon - \mathbf{C}'\|_1 \leq \epsilon$ . By triangle inequality, this implies that

$$\|\mathbf{C}_\epsilon - \mathbf{C}^*\|_1 \geq \xi - \epsilon , \tag{3}$$

Let  $\mathbf{f}_\epsilon : \Delta_n \rightarrow \Delta_n$  be s.t.  $\mathbf{C}_\epsilon = \mathbf{C}^D[\mathbf{f}_\epsilon \circ \eta]$ . Now we describe the set of conditional probabilities  $\mathbf{p} \in \Delta_n$  for which  $\mathbf{f}_\epsilon(\mathbf{p})$  differs significantly from  $\mathbf{f}^*(\mathbf{p})$ . Denote this ‘bad’ set as  $\mathcal{B}$ . Let

$$\mathcal{B} = \{\mathbf{p} \in \Delta_n : \|\mathbf{f}^*(\mathbf{p}) - \mathbf{f}_\epsilon(\mathbf{p})\|_1 \geq \frac{\xi}{4}\} .$$

We now show that this set is ‘large’. Applying Eq. 3 and Lemma 21 we have

$$\begin{aligned} \xi - \epsilon &\leq \|\mathbf{C}^D[\mathbf{f}^* \circ \eta] - \mathbf{C}^D[\mathbf{f}_\epsilon \circ \eta]\|_1 \\ &\leq \int_{\mathbf{p} \in \Delta_n} \|\mathbf{f}^*(\mathbf{p}) - \mathbf{f}_\epsilon(\mathbf{p})\|_1 d\nu(\mathbf{p}) \\ &\leq \int_{\mathbf{p} \in \mathcal{B}} 2 d\nu(\mathbf{p}) + \int_{\mathbf{p} \notin \mathcal{B}} \frac{\xi}{4} d\nu(\mathbf{p}) \\ &= 2\nu(\mathcal{B}) + \frac{\xi}{4}(1 - \nu(\mathcal{B})) \\ &\leq 2\nu(\mathcal{B}) + \frac{\xi}{4} \\ \nu(\mathcal{B}) &\geq \frac{3\xi}{8} - \frac{\epsilon}{2} \end{aligned} \tag{4}$$

Now we consider the set of conditional probabilities  $\mathbf{p} \in \Delta_n$ , such that the second best and best predictions (in terms of  $\mathbf{L}$ ) are ‘close’ in performance. We show that this set is ‘small’.

For any  $c > 0$ , define  $\mathcal{A}_c \subseteq \Delta_n$  as

$$\mathcal{A}_c = \{\mathbf{p} \in \Delta_n : (\mathbf{p}^\top \mathbf{L})_{(2)} - (\mathbf{p}^\top \mathbf{L})_{(1)} \leq c\}.$$

From Lemma 23 we have that  $\nu(\mathcal{A}_c)$  is a continuous function of  $c$  close to 0 and  $\nu(\mathcal{A}_0) = 0$ . Let  $c > 0$  be such that

$$\nu(\mathcal{A}_c) \leq \frac{\xi}{16}. \quad (5)$$

From Eq. 4 and 5, we have

$$\nu(\mathcal{B} \setminus \mathcal{A}_c) \geq \frac{5\xi}{16} - \frac{\epsilon}{2}$$

Any  $\mathbf{p} \in \mathcal{B} \setminus \mathcal{A}_c$  is such that  $\mathbf{f}_\epsilon(\mathbf{p})$  is different from  $\mathbf{f}^*(\mathbf{p})$ , and the second best prediction is significantly worse than the best prediction, i.e.

$$(\mathbf{p}^\top \mathbf{L})_{(2)} - (\mathbf{p}^\top \mathbf{L})_{(1)} > c \quad \text{and} \quad \|\mathbf{f}^*(\mathbf{p}) - \mathbf{f}_\epsilon(\mathbf{p})\|_1 \geq \frac{\xi}{4}.$$

For any  $\mathbf{p} \in \Delta_n$ , we have  $\mathbf{f}^*(\mathbf{p}) \in \Delta_n$ , has a 1 at the index  $\operatorname{argmin}_{y \in [n]} \mathbf{p}^\top \ell_y$  and zero elsewhere. For any  $\mathbf{p} \in \mathcal{B} \setminus \mathcal{A}_c$ , we have  $\|\mathbf{f}^*(\mathbf{p}) - \mathbf{f}_\epsilon(\mathbf{p})\|_1 \geq \frac{\xi}{4}$ , and hence the value of  $\mathbf{f}_\epsilon(\mathbf{p})$  corresponding to the index  $\operatorname{argmin}_{y \in [n]} \mathbf{p}^\top \ell_y$ , is at most  $(1 - \frac{\xi}{8})$ . In particular, we have

$$\mathbf{p}^\top \mathbf{L} \mathbf{f}_\epsilon(\mathbf{p}) \geq \left(1 - \frac{\xi}{8}\right) (\mathbf{p}^\top \mathbf{L})_{(1)} + \left(\frac{\xi}{8}\right) (\mathbf{p}^\top \mathbf{L})_{(2)}. \quad (6)$$

By using Lemma 19 and Eq. 6 we have,

$$\begin{aligned} \langle \mathbf{L}, \mathbf{C}_\epsilon \rangle - \langle \mathbf{L}, \mathbf{C}^* \rangle &= \int_{\mathbf{p} \in \Delta_n} \mathbf{p}^\top \mathbf{L} (\mathbf{f}_\epsilon(\mathbf{p}) - \mathbf{f}^*(\mathbf{p})) d\nu(\mathbf{p}) \\ &= \int_{\mathbf{p} \in \mathcal{B} \setminus \mathcal{A}_c} \mathbf{p}^\top \mathbf{L} (\mathbf{f}_\epsilon(\mathbf{p}) - \mathbf{f}^*(\mathbf{p})) d\nu(\mathbf{p}) + \int_{\mathbf{p} \in \Delta_n \setminus (\mathcal{B} \setminus \mathcal{A}_c)} \mathbf{p}^\top \mathbf{L} (\mathbf{f}_\epsilon(\mathbf{p}) - \mathbf{f}^*(\mathbf{p})) d\nu(\mathbf{p}) \\ &\geq \int_{\mathbf{p} \in \mathcal{B} \setminus \mathcal{A}_c} \mathbf{p}^\top \mathbf{L} (\mathbf{f}_\epsilon(\mathbf{p}) - \mathbf{f}^*(\mathbf{p})) d\nu(\mathbf{p}) \\ &\geq \int_{\mathbf{p} \in \mathcal{B} \setminus \mathcal{A}_c} \left( \left(1 - \frac{\xi}{8}\right) (\mathbf{p}^\top \mathbf{L})_{(1)} + \left(\frac{\xi}{8}\right) (\mathbf{p}^\top \mathbf{L})_{(2)} - (\mathbf{p}^\top \mathbf{L})_{(1)} \right) d\nu(\mathbf{p}) \\ &= \int_{\mathbf{p} \in \mathcal{B} \setminus \mathcal{A}_c} \frac{\xi}{8} ((\mathbf{p}^\top \mathbf{L})_{(2)} - (\mathbf{p}^\top \mathbf{L})_{(1)}) d\nu(\mathbf{p}) \\ &\geq \frac{\xi c}{8} \left( \frac{5\xi}{16} - \frac{\epsilon}{2} \right) \end{aligned}$$

If  $\epsilon \leq \frac{\xi}{2}$ , we have

$$\langle \mathbf{L}, \mathbf{C}_\epsilon \rangle - \langle \mathbf{L}, \mathbf{C}^* \rangle \geq \frac{\xi^2 c}{128}. \quad (7)$$

The above holds for any  $\epsilon \in (0, \frac{\xi}{2}]$ , and both  $\xi$  and  $c$  do not depend on  $\epsilon$ .

However, we have  $\mathbf{C}' \in \operatorname{argmin}_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{L}, \mathbf{C} \rangle$ , and  $\|\mathbf{C}_\epsilon - \mathbf{C}'\|_1 \leq \epsilon$ . Hence,

$$\begin{aligned} \langle \mathbf{L}, \mathbf{C}_\epsilon \rangle &= \langle \mathbf{L}, \mathbf{C}' \rangle + \langle \mathbf{L}, \mathbf{C}_\epsilon - \mathbf{C}' \rangle \\ &\leq \langle \mathbf{L}, \mathbf{C}' \rangle + \|\mathbf{L}\|_\infty \|\mathbf{C}_\epsilon - \mathbf{C}'\|_1 \end{aligned}$$

$$\begin{aligned}
 &\leq \langle \mathbf{L}, \mathbf{C}' \rangle + \epsilon \\
 &\leq \min_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{L}, \mathbf{C} \rangle + \epsilon \\
 &= \langle \mathbf{L}, \mathbf{C}^* \rangle + \epsilon
 \end{aligned} \tag{8}$$

It can be clearly seen that, for small enough  $\epsilon$ , Eqs. 7 and 8 contradict each other. And thus we have that  $\mathbf{C}' = \mathbf{C}^*$ .  $\square$

### B.5. Proof of Lemma 12

**Lemma** (Existence of Bayes optimal classifier for monotonic  $\psi$ ). *Let  $D$  be such that the probability measure associated with the random vector  $\boldsymbol{\eta}(X) = (\eta_1(X), \dots, \eta_n(X))^\top$  is absolutely continuous w.r.t. the base probability measure associated with the uniform distribution over  $\Delta_n$ , and let  $\psi$  be a performance measure that is differentiable and bounded over  $\mathcal{C}_D$ , and is monotonically increasing in  $C_{ii}$  for each  $i$  and non-increasing in  $C_{ij}$  for all  $i, j$ . Then  $\exists h^* : \mathcal{X} \rightarrow \Delta_n$  s.t.  $\mathcal{P}_D^\psi[h^*] = \mathcal{P}_D^{\psi,*}$ .*

*Proof.* Let  $\mathbf{C}^* = \operatorname{argmax}_{\mathbf{C} \in \overline{\mathcal{C}_D}} \psi(\mathbf{C})$ . Such a  $\mathbf{C}^*$  always exists by compactness of  $\overline{\mathcal{C}_D}$  and continuity of  $\psi$ . We will show that this  $\mathbf{C}^*$  is also in  $\mathcal{C}_D$ , thus proving the existence of  $h^* : \mathcal{X} \rightarrow \Delta_n$  which is such that  $\mathbf{C}^* = \mathbf{C}^D[h^*]$  and hence  $\mathcal{P}_D^\psi[h^*] = \mathcal{P}_D^{\psi,*}$ .

By first order optimality, and convexity of  $\overline{\mathcal{C}_D}$ , we have that for all  $\mathbf{C} \in \overline{\mathcal{C}_D}$

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

Let  $\mathbf{L}^*$  be the scaled and shifted version of  $-\nabla \psi(\mathbf{C}^*)$  with entries in  $[0, 1]$ , then we have that

$$\mathbf{C}^* \in \operatorname{argmin}_{\mathbf{C} \in \overline{\mathcal{C}_D}} \langle \mathbf{L}^*, \mathbf{C} \rangle.$$

Due to the monotonicity condition on  $\psi$  the diagonal elements of its gradient  $\nabla \psi(\mathbf{C}^*)$  are positive, and the off-diagonal elements are non-positive, and hence no two columns of  $\mathbf{L}^*$  are identical. Thus by Lemma 25, we have that  $\mathbf{C}^* \in \mathcal{C}_D$ .  $\square$

### B.6. Proof of Theorem 13

**Theorem** (Form of Bayes optimal classifier for monotonic  $\psi$ ). *Let  $D, \psi$  satisfy the conditions of Lemma 12. Let  $h^* : \mathcal{X} \rightarrow \Delta_n$  be a  $\psi$ -optimal classifier and let  $\mathbf{C}^* = \mathbf{C}^D[h^*]$ . Let  $\tilde{\mathbf{L}}^* = -\nabla \psi(\mathbf{C}^*)$ , and let  $\mathbf{L}^* \in [0, 1]^{n \times n}$  be obtained by scaling and shifting  $\tilde{\mathbf{L}}^*$  so its entries lie in  $[0, 1]$ . Then any classifier that is  $\psi^{\mathbf{L}^*}$ -optimal is also  $\psi$ -optimal.*

*Proof.* Clearly

$$\psi(\mathbf{C}^*) = \max_{\mathbf{C} \in \mathcal{C}_D} \psi(\mathbf{C}).$$

Hence by the differentiability of  $\psi$ , first order conditions for optimality and convexity of  $\mathcal{C}_D$  we have  $\forall \mathbf{C} \in \mathcal{C}_D$ ,

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

By definition of  $\mathbf{L}^*$ , this implies that  $\forall \mathbf{C} \in \mathcal{C}_D$ ,

$$\langle \mathbf{L}^*, \mathbf{C}^* \rangle \leq \langle \mathbf{L}^*, \mathbf{C} \rangle.$$

Thus, we have that  $h^*$  is a  $\psi^{\mathbf{L}^*}$ -optimal classifier.

Due to the monotonicity condition on  $\psi$  the diagonal elements of its gradient  $\nabla \psi(\mathbf{C}^*)$  are positive, and the off-diagonal elements are non-positive, and hence no two columns of  $\mathbf{L}^*$  are identical. By Lemma 24 (or Lemma 25), we have that all  $\psi^{\mathbf{L}^*}$ -optimal classifiers have the same confusion matrix, which is equal to  $\mathbf{C}^D[h^*] = \mathbf{C}^*$ . And thus all  $\psi^{\mathbf{L}^*}$  optimal classifiers are also  $\psi$ -optimal.  $\square$

## C. Supplementary Material for Section 5 (Consistency)

### C.1. Proof of Lemma 14

**Lemma** (**L**-regret of multiclass plug-in classifiers). *For a fixed  $\mathbf{L} \in [0, 1]^{n \times n}$  and class probability estimation model  $\hat{\boldsymbol{\eta}} : \mathcal{X} \rightarrow \Delta_n$ , let  $\hat{h} : \mathcal{X} \rightarrow [n]$  be a classifier  $\hat{h}(x) = \operatorname{argmin}_{j \in [n]} \sum_{i=1}^n \hat{\eta}_i(x) L_{ij}$ . Then*

$$\mathcal{P}_D^{\mathbf{L},*} - \mathcal{P}_D^{\mathbf{L}}[\hat{h}] \leq \mathbf{E}_X [\|\hat{\boldsymbol{\eta}}(X) - \boldsymbol{\eta}(X)\|_1].$$

*Proof.* Let  $h^* : \mathcal{X} \rightarrow \Delta_n$  be such that

$$h^*(x) = \operatorname{argmin}_{y \in [n]} \boldsymbol{\ell}_y^\top \boldsymbol{\eta}(x).$$

By Proposition 6, we have that

$$h^* \in \operatorname{argmax}_{h : \mathcal{X} \rightarrow \Delta_n} \mathcal{P}_D^{\mathbf{L}}[h].$$

We then have

$$\begin{aligned} \mathcal{P}_D^{\mathbf{L},*} - \mathcal{P}_D^{\mathbf{L}}[\hat{h}] &= \langle \mathbf{L}, \mathbf{C}^D[\hat{h}] \rangle - \langle \mathbf{L}, \mathbf{C}^D[h^*] \rangle \\ &= \mathbf{E}_X [\boldsymbol{\eta}(X)^\top \boldsymbol{\ell}_{\hat{h}(X)}] - \mathbf{E}_X [\boldsymbol{\eta}(X)^\top \boldsymbol{\ell}_{h^*(X)}] \\ &= \mathbf{E}_X [\hat{\boldsymbol{\eta}}(X)^\top \boldsymbol{\ell}_{\hat{h}(X)}] + \mathbf{E}_X [(\boldsymbol{\eta}(X) - \hat{\boldsymbol{\eta}}(X))^\top \boldsymbol{\ell}_{\hat{h}(X)}] - \mathbf{E}_X [\boldsymbol{\eta}(X)^\top \boldsymbol{\ell}_{h^*(X)}] \\ &\leq \mathbf{E}_X [\hat{\boldsymbol{\eta}}(X)^\top \boldsymbol{\ell}_{h^*(X)}] + \mathbf{E}_X [(\boldsymbol{\eta}(X) - \hat{\boldsymbol{\eta}}(X))^\top \boldsymbol{\ell}_{\hat{h}(X)}] - \mathbf{E}_X [\boldsymbol{\eta}(X)^\top \boldsymbol{\ell}_{h^*(X)}] \\ &= \mathbf{E}_X [(\boldsymbol{\eta}(X) - \hat{\boldsymbol{\eta}}(X))^\top (\boldsymbol{\ell}_{\hat{h}(X)} - \boldsymbol{\ell}_{h^*(X)})] \\ &\leq \mathbf{E}_X [\|\boldsymbol{\eta}(X) - \hat{\boldsymbol{\eta}}(X)\|_1 \cdot \|\boldsymbol{\ell}_{\hat{h}(X)} - \boldsymbol{\ell}_{h^*(X)}\|_\infty] \\ &\leq \mathbf{E}_X [\|\boldsymbol{\eta}(X) - \hat{\boldsymbol{\eta}}(X)\|_1], \end{aligned}$$

as desired.  $\square$

### C.2. Proof of Lemma 15

**Lemma** (Uniform convergence of confusion matrices). *For  $\mathbf{q} : \mathcal{X} \rightarrow \Delta_n$ , let*

$$\mathcal{H}_{\mathbf{q}} = \{h : \mathcal{X} \rightarrow [n], h(x) = \operatorname{argmin}_{j \in [n]} \sum_{i=1}^n q_i(x) L_{ij} \mid \mathbf{L} \in [0, 1]^{n \times n}\}.$$

*Let  $S \in (\mathcal{X} \times [n])^m$  be a sample drawn i.i.d. from  $D^m$ . For any  $\delta \in (0, 1]$ , w.p. at least  $1 - \delta$  (over draw of  $S$  from  $D^m$ ),*

$$\sup_{h \in \mathcal{H}_{\mathbf{q}}} \|\mathbf{C}^D[h] - \hat{\mathbf{C}}^S[h]\|_\infty \leq C \sqrt{\frac{n^2 \log(n) \log(m) + \log(n^2/\delta)}{m}},$$

where  $C > 0$  is a distribution-independent constant.

*Proof.* For any  $a, b \in [n]$  we have,

$$\begin{aligned} \sup_{h \in \mathcal{H}_{\mathbf{q}}} \left| \hat{C}_{a,b}^S[h] - C_{a,b}^D[h] \right| &= \sup_{h \in \mathcal{H}_{\mathbf{q}}} \left| \frac{1}{m} \sum_{i=1}^m (\mathbf{1}(y_i = a, h(x_i) = b) - \mathbf{E}[\mathbf{1}(Y = a, h(X) = b)]) \right| \\ &= \sup_{h \in \mathcal{H}_{\mathbf{q}}^b} \left| \frac{1}{m} \sum_{i=1}^m (\mathbf{1}(y_i = a, h(x_i) = 1) - \mathbf{E}[\mathbf{1}(Y = a, h(X) = 1)]) \right|, \end{aligned}$$

where for a fixed  $b \in [n]$ ,  $\mathcal{H}_{\mathbf{q}}^b = \{h : \mathcal{X} \rightarrow \{0, 1\} : \exists \mathbf{L} \in [0, 1]^{n \times n}, \forall x \in \mathcal{X}, h(x) = \mathbf{1}(b = \operatorname{argmin}_{y \in [n]} \boldsymbol{\ell}_y^\top \mathbf{q}(x))\}$ . The set  $\mathcal{H}_{\mathbf{q}}^b$  can be seen as hypothesis class whose concepts are the intersection of  $n$  halfspaces in  $\mathbb{R}^n$  (corresponding to  $\mathbf{q}(x)$ )



through the origin. Hence we have from Lemma 3.2.3 of Blumer et al. (1989) that the VC-dimension of  $\mathcal{H}_{\mathbf{q}}^b$  is at most  $2n^2 \log(3n)$ . From standard uniform convergence arguments we have that for each  $a, b \in [n]$ , the following holds with at least probability  $1 - \delta$ ,

$$\sup_{h \in \mathcal{H}_{\mathbf{q}}} \left| \widehat{C}_{a,b}^S[h] - C_{a,b}^D[h] \right| \leq C \sqrt{\frac{n^2 \log(n) \log(m) + \log(\frac{1}{\delta})}{m}}$$

where  $C > 0$  is some constant. Applying union bound over all  $a, b \in [n]$  we have that the following holds with probability at least  $1 - \delta$

$$\sup_{h \in \mathcal{H}_{\mathbf{q}}} \left\| \widehat{\mathbf{C}}^S[h] - \mathbf{C}^D[h] \right\|_{\infty} \leq C \sqrt{\frac{n^2 \log(n) \log(m) + \log(\frac{n^2}{\delta})}{m}}.$$

□

### C.3. Proof of Theorem 16

**Theorem** ( $\psi$ -regret of Frank-Wolfe method based algorithm). *Let  $\psi : [0, 1]^{n \times n} \rightarrow \mathbb{R}_+$  be concave over  $\mathcal{C}_D$ , and  $L$ -Lipschitz and  $\beta$ -smooth w.r.t. the  $\ell_1$  norm. Let  $S = (S_1, S_2) \in (\mathcal{X} \times [n])^m$  be a training sample drawn i.i.d. from  $D$ . Further, let  $\widehat{\eta} : \mathcal{X} \rightarrow \Delta_n$  be the CPE model learned from  $S_1$  in Algorithm 1 and  $h_S^{\text{FW}} : \mathcal{X} \rightarrow \Delta_n$  be the classifier obtained after  $\kappa m$  iterations. Then for any  $\delta \in (0, 1]$ , with probability at least  $1 - \delta$  (over draw of  $S$  from  $D^m$ ),*

$$\mathcal{P}_D^{\psi,*} - \mathcal{P}_D^{\psi}[h_S^{\text{FW}}] \leq 4L\mathbf{E}_X[\|\widehat{\eta}(X) - \eta(X)\|_1] + 4\sqrt{2}\beta n^2 C \sqrt{\frac{n^2 \log(n) \log(m) + \log(n^2/\delta)}{m}} + \frac{8\beta}{\kappa m + 2},$$

where  $C > 0$  is a distribution-independent constant.

We first prove an important lemma where we bound the approximation error of the linear optimization oracle used in the algorithm using Lemma 14 and 15. This result coupled with the standard convergence analysis for the Frank-Wolfe method (Jaggi, 2013) will then allow us to prove the above theorem.

**Lemma 26.** *Let  $\psi : [0, 1]^{n \times n} \rightarrow \mathbb{R}_+$  be concave over  $\mathcal{C}_D$ , and  $L$ -Lipschitz and  $\beta$ -smooth w.r.t. the  $\ell_1$  norm. Let classifiers  $\widehat{g}^1, \dots, \widehat{g}^T$ , and  $h^0, h^1, \dots, h^T$  be as defined in Algorithm 1. Then for any  $\delta \in (0, 1]$ , with probability at least  $1 - \delta$  (over draw of  $S$  from  $D^m$ ), we have for all  $1 \leq t \leq T$*

$$\langle \nabla \psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C}^D[\widehat{g}^t] \rangle \geq \max_{\mathbf{g} : \mathcal{X} \rightarrow \Delta_n} \langle \nabla \psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C}^D[\mathbf{g}] \rangle - \epsilon_S$$

where

$$\epsilon_S = 2L\mathbf{E}_X[\|\eta(X) - \widehat{\eta}(X)\|_1] + 2\sqrt{2}C\beta n^2 \sqrt{\frac{n^2 \log(n) \log(m) + \log(\frac{n^2}{\delta})}{m}}.$$

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

$$\begin{aligned} & \max_{\mathbf{g} : \mathcal{X} \rightarrow \Delta_n} \langle \nabla \psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C}^D[\mathbf{g}] \rangle - \langle \nabla \psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C}^D[\widehat{g}^t] \rangle \\ &= \langle \nabla \psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C}^D[\mathbf{g}^{t,*}] \rangle - \langle \nabla \psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C}^D[\widehat{g}^t] \rangle \\ &= \underbrace{\langle \nabla \psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C}^D[\mathbf{g}^{t,*}] \rangle - \langle \nabla \psi(\widehat{\mathbf{C}}^{S_2}[h^{t-1}]), \mathbf{C}^D[\mathbf{g}^{t,*}] \rangle}_{\text{term}_1} \\ & \quad + \underbrace{\langle \nabla \psi(\widehat{\mathbf{C}}^{S_2}[h^{t-1}]), \mathbf{C}^D[\mathbf{g}^{t,*}] \rangle - \langle \nabla \psi(\widehat{\mathbf{C}}^{S_2}[h^{t-1}]), \mathbf{C}^D[\widehat{g}^t] \rangle}_{\text{term}_2} \\ & \quad + \underbrace{\langle \nabla \psi(\widehat{\mathbf{C}}^{S_2}[h^{t-1}]), \mathbf{C}^D[\widehat{g}^t] \rangle - \langle \nabla \psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C}^D[\widehat{g}^t] \rangle}_{\text{term}_3}. \end{aligned}$$

We next bound each of these terms. We start with term<sub>2</sub>. For any  $1 \leq t \leq T$ , let  $\hat{\mathbf{L}}^t$  be as defined in Algorithm 1. Since  $\hat{\mathbf{L}}^t$  is a scaled and translated version of the gradient  $\nabla\psi(\hat{\mathbf{C}}^{S_2}[h^{t-1}])$ , we have  $\nabla\psi(\hat{\mathbf{C}}^{S_2}[h^{t-1}]) = c_t - a_t \hat{\mathbf{L}}^t$ , for some constant  $c_t \in \mathbb{R}$  and  $a_t \in [0, 2L]$ . Thus for all  $1 \leq t \leq T$ ,

$$\begin{aligned} \langle \nabla\psi(\hat{\mathbf{C}}^{S_2}[h^{t-1}]), \mathbf{C}^D[\mathbf{g}^{t,*}] \rangle - \langle \nabla\psi(\hat{\mathbf{C}}^{S_2}[h^{t-1}]), \mathbf{C}^D[\hat{\mathbf{g}}^t] \rangle \\ &= a_t \cdot (\langle \hat{\mathbf{L}}^t, \mathbf{C}^D[\hat{\mathbf{g}}^t] \rangle - \langle \hat{\mathbf{L}}^t, \mathbf{C}^D[\mathbf{g}^{t,*}] \rangle) \\ &= a_t \cdot (\mathcal{P}_D^{\hat{\mathbf{L}}^t}[\mathbf{g}^{t,*}] - \mathcal{P}_D^{\hat{\mathbf{L}}^t}[\hat{\mathbf{g}}^t]) \\ &\leq a_t \cdot (\mathcal{P}_D^{\hat{\mathbf{L}}^t,*} - \mathcal{P}_D^{\hat{\mathbf{L}}^t}[\hat{\mathbf{g}}^t]) \\ &\leq 2L \cdot (\mathcal{P}_D^{\hat{\mathbf{L}}^t,*} - \mathcal{P}_D^{\hat{\mathbf{L}}^t}[\hat{\mathbf{g}}^t]) \\ &\leq 2L \cdot \mathbf{E}_X [\|\boldsymbol{\eta}(X) - \hat{\boldsymbol{\eta}}(X)\|_1], \end{aligned}$$

where the third step uses the definition of  $\mathcal{P}_D^{\hat{\mathbf{L}}^t,*}$  and the last step follows from Lemma 14.

Next, for term<sub>1</sub>, we have by an application of Holder's inequality

$$\begin{aligned} \langle \nabla\psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C}^D[\mathbf{g}^{t,*}] \rangle - \langle \nabla\psi(\hat{\mathbf{C}}^{S_2}[h^{t-1}]), \mathbf{C}^D[\mathbf{g}^{t,*}] \rangle \\ &\leq \|\nabla\psi(\hat{\mathbf{C}}^{S_2}[h^{t-1}]) - \nabla\psi(\mathbf{C}^D[h^{t-1}])\|_\infty \|\mathbf{C}^D[\mathbf{g}^{t,*}]\|_1 \\ &= \|\nabla\psi(\hat{\mathbf{C}}^{S_2}[h^{t-1}]) - \nabla\psi(\mathbf{C}^D[h^{t-1}])\|_\infty (1) \\ &\leq \beta \|\hat{\mathbf{C}}^{S_2}[h^{t-1}] - \mathbf{C}^D[h^{t-1}]\|_1 \\ &\leq \beta n^2 \|\hat{\mathbf{C}}^{S_2}[h^{t-1}] - \mathbf{C}^D[h^{t-1}]\|_\infty \\ &\leq \beta n^2 \max_{i \in [t-1]} \|\hat{\mathbf{C}}^{S_2}[\hat{\mathbf{g}}^i] - \mathbf{C}^D[\hat{\mathbf{g}}^i]\|_\infty \\ &\leq \beta n^2 \sup_{h \in \mathcal{H}_{\hat{\boldsymbol{\eta}}}} \|\hat{\mathbf{C}}^{S_2}[h] - \mathbf{C}^D[h]\|_\infty, \end{aligned}$$

where the third step follows from  $\beta$ -smoothness of  $\psi$ . One can similarly bound term<sub>3</sub>. We thus have for all  $1 \leq t \leq T$ ,

$$\begin{aligned} \max_{\mathbf{g}: \mathcal{X} \rightarrow \Delta_n} \langle \nabla\psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C}^D[\mathbf{g}] \rangle - \langle \nabla\psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C}^D[\hat{\mathbf{g}}^t] \rangle \\ \leq 2L \cdot \mathbf{E}_X [\|\boldsymbol{\eta}(X) - \hat{\boldsymbol{\eta}}(X)\|_1] + 2\beta n^2 \sup_{h \in \mathcal{H}_{\hat{\boldsymbol{\eta}}}} \|\hat{\mathbf{C}}^{S_2}[h] - \mathbf{C}^D[h]\|_\infty. \end{aligned}$$

Applying Lemma 15 with  $|S_2| = \lceil m/2 \rceil$  examples, we have with probability  $1 - \delta$  (over random draw of  $S_2$  from  $D$ ), for all  $1 \leq t \leq T$ ,

$$\begin{aligned} \max_{\mathbf{g}: \mathcal{X} \rightarrow \Delta_n} \langle \nabla\psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C}^D[\mathbf{g}] \rangle - \langle \nabla\psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C}^D[\hat{\mathbf{g}}^t] \rangle \\ \leq 2L \mathbf{E}_X [\|\boldsymbol{\eta}(X) - \hat{\boldsymbol{\eta}}(X)\|_1] + 2\sqrt{2}C\beta n^2 \sqrt{\frac{n^2 \log(n) \log(m) + \log(\frac{n^2}{\delta})}{m}}. \end{aligned}$$

□

We are now ready to prove Theorem 16.

*Proof of Theorem 16.* Our proof shall make use of Lemma 26 and the standard convergence result for the Frank-Wolfe algorithm for maximizing a concave function over a convex set (Jaggi, 2013). We will find it useful to first define the following quantity, referred to as the curvature constant in (Jaggi, 2013).

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

Also, define two positive scalars  $\epsilon_S$  and  $\delta_{\text{apx}}$  required in the analysis of (Jaggi, 2013):

$$\epsilon_S = 2L \mathbf{E}_X [\|\boldsymbol{\eta}(X) - \hat{\boldsymbol{\eta}}(X)\|_1] + 2\sqrt{2}C\beta n^2 \sqrt{\frac{n^2 \log(n) \log(m) + \log(\frac{n^2}{\delta})}{m}}$$

$$\delta_{\text{apx}} = \frac{(T+1)\epsilon_S}{C_\psi},$$

where  $\delta \in (0, 1]$  is as in the theorem statement. Further, let the classifiers  $\hat{g}^1, \dots, \hat{g}^T$ , and  $h^0, h^1, \dots, h^T$  be as defined in Algorithm 1. We then have from Lemma 26 that the following holds with probability at least  $1 - \delta$ , for all  $1 \leq t \leq T$ ,

$$\begin{aligned} \langle \nabla \psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C}^D[\hat{g}^t] \rangle &\geq \max_{\mathbf{g}: \mathcal{X} \rightarrow \Delta_n} \langle \nabla \psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C}^D[\mathbf{g}] \rangle - \epsilon_S \\ &= \max_{\mathbf{C} \in \mathcal{C}_D} \langle \nabla \psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C} \rangle - \epsilon_S \\ &= \max_{\mathbf{C} \in \bar{\mathcal{C}}_D} \langle \nabla \psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C} \rangle - \epsilon_S \\ &= \max_{\mathbf{C} \in \bar{\mathcal{C}}_D} \langle \nabla \psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C} \rangle - \frac{1}{2} \delta_{\text{apx}} \frac{2}{T+1} C_\psi \\ &\geq \max_{\mathbf{C} \in \bar{\mathcal{C}}_D} \langle \nabla \psi(\mathbf{C}^D[h^{t-1}]), \mathbf{C} \rangle - \frac{1}{2} \delta_{\text{apx}} \frac{2}{t+1} C_\psi. \end{aligned} \quad (9)$$

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

$$\mathbf{C}^D[h^t] = \left(1 - \frac{2}{t+1}\right) \mathbf{C}^D[h^{t-1}] + \frac{2}{t+1} \mathbf{C}^D[\hat{g}^t], \quad (10)$$

for all  $1 \leq t \leq T$ . Based on Eq. (9) and Eq. (10), one can now apply the result of (Jaggi, 2013).

In particular, the sequence of iterates  $\mathbf{C}^D[h^0], \mathbf{C}^D[h^1], \dots, \mathbf{C}^D[h^T]$  can be considered as the sequence of iterates arising from running the Frank-Wolfe optimization method to maximize  $\psi$  over  $\bar{\mathcal{C}}_D$  with a linear optimization oracle that is  $\frac{1}{2} \delta_{\text{apx}} \frac{2}{t+1} C_\psi$  accurate at iteration  $t$ . Since  $\psi$  is a concave function over the convex constraint set  $\bar{\mathcal{C}}_D$ , one has from Theorem 1 in (Jaggi, 2013) that the following convergence guarantee holds with probability at least  $1 - \delta$ :

$$\begin{aligned} \mathcal{P}_D^\psi[h_S^{\text{FW}}] &= \psi(\mathbf{C}^D[h_S^{\text{FW}}]) \\ &= \psi(\mathbf{C}^D[h^T]) \\ &\geq \max_{\mathbf{C} \in \bar{\mathcal{C}}_D} \psi(\mathbf{C}) - \frac{2C_\psi}{T+2} (1 + \delta_{\text{apx}}) \\ &= \max_{\mathbf{C} \in \bar{\mathcal{C}}_D} \psi(\mathbf{C}) - \frac{2C_\psi}{T+2} - \frac{2\epsilon_S(T+1)}{T+2} \\ &\geq \max_{\mathbf{C} \in \bar{\mathcal{C}}_D} \psi(\mathbf{C}) - \frac{2C_\psi}{T+2} - 2\epsilon_S \end{aligned} \quad (11)$$

We can further upper bound  $C_\psi$  in the above inequality in terms of the smoothness parameter of  $\psi$ :

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

where the second step follows from the  $\beta$ -smoothness of  $\psi$ . Substituting back in Eq. (11), we finally have with probability at least  $1 - \delta$ ,

$$\begin{aligned} \mathcal{P}_D^\psi[h_S^{\text{FW}}] &\geq \max_{\mathbf{C} \in \bar{\mathcal{C}}_D} \psi(\mathbf{C}) - \frac{8\beta}{T+2} - 2\epsilon_S \\ &= \mathcal{P}_D^{\psi, *} - 4L\mathbf{E}_X[\|\boldsymbol{\eta}(X) - \hat{\boldsymbol{\eta}}(X)\|_1] - 4\sqrt{2}C\beta n^2 \sqrt{\frac{n^2 \log(n) \log(m) + \log(\frac{n^2}{\delta})}{m}} - \frac{8\beta}{T+2}, \end{aligned}$$

which follows from the definition of  $\epsilon_S$ . Setting  $T = \kappa m$  completes the proof.  $\square$

#### C.4. Proof of Theorem 17

**Theorem.** Let  $\psi : [0, 1]^{n \times n} \rightarrow \mathbb{R}_+$  be such that  $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$  where  $\mathbf{A}, \mathbf{B} \in \mathbb{R}_+^{n \times n}$ ,  $\sup_{\mathbf{C} \in \mathcal{C}_D} \psi(\mathbf{C}) \leq 1$ , and  $\min_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{B}, \mathbf{C} \rangle \geq b$ , for some  $b > 0$ . Let  $S = (S_1, S_2) \in (\mathcal{X} \times [n])^m$  be a training sample drawn i.i.d. from  $D$ . Let  $\hat{\eta} : \mathcal{X} \rightarrow \Delta_n$  be the CPE model learned from  $S_1$  in Algorithm 2 and  $h_S^{\text{BS}} : \mathcal{X} \rightarrow [n]$  be the classifier obtained after  $\kappa m$  iterations. Then for any  $\delta \in (0, 1]$ , with probability at least  $1 - \delta$  (over draw of  $S$  from  $D^m$ ),

$$\mathcal{P}_D^* - \mathcal{P}_D[h_S^{\text{BS}}] \leq 2\tau \mathbf{E}_X[\|\hat{\eta}(X) - \eta(X)\|_1] + 2\sqrt{2}C\tau \sqrt{\frac{n^2 \log(n) \log(m) + \log(n^2/\delta)}{m}} + 2^{-\kappa m},$$

where  $\tau = \frac{1}{b}(\|\mathbf{A}\|_1 + \|\mathbf{B}\|_1)$  and  $C > 0$  is a distribution-independent constant.

We will find it useful to state the following lemmas:

**Lemma 27 (Invariant in Algorithm 2).** Let  $\psi$  be as defined in Theorem 17. Let  $\mathcal{H}_{\hat{\eta}} = \{h : \mathcal{X} \rightarrow [n], h(x) = \arg\min_{j \in [n]} \sum_{i=1}^n \hat{\eta}_i(x) L_{ij} \mid \mathbf{L} \in [0, 1]^{n \times n}\}$ . Then the following invariant is true at the end of each iteration  $0 \leq t \leq T$  of Algorithm 2:

$$\alpha^t - \tau \bar{\epsilon} < \psi(\mathbf{C}^D[h^t]) \leq \sup_{\mathbf{C} \in \mathcal{C}_D} \psi(\mathbf{C}) \leq \beta^t + \tau \bar{\epsilon},$$

where  $\tau = \frac{1}{b}(\|\mathbf{A}\|_1 + \|\mathbf{B}\|_1)$  and  $\bar{\epsilon} = \mathbf{E}_X[\|\hat{\eta}(X) - \eta(X)\|_1] + \sup_{h \in \mathcal{H}_{\hat{\eta}}} \|\mathbf{C}^D[h] - \hat{\mathbf{C}}^{S_2}[h]\|_\infty$ .

*Proof.* We first have from Lemma 14, the following guarantee for the linear minimization step at each iteration  $t$  of Algorithm 2:

$$\begin{aligned} \langle \hat{\mathbf{L}}^t, \mathbf{C}^D[\hat{g}^t] \rangle &\leq \min_{\mathbf{C} \in \mathcal{C}_D} \langle \hat{\mathbf{L}}^t, \mathbf{C} \rangle + \mathbf{E}_X[\|\hat{\eta}(X) - \eta(X)\|_1] \\ &= \min_{\mathbf{C} \in \mathcal{C}_D} \langle \hat{\mathbf{L}}^t, \mathbf{C} \rangle + \epsilon \quad (\text{say}). \end{aligned} \tag{12}$$

Further, let us denote  $\epsilon' = \sup_{h \in \mathcal{H}_{\hat{\eta}}} \|\mathbf{C}^D[h] - \hat{\mathbf{C}}^{S_2}[h]\|_\infty$ . Notice that  $\bar{\epsilon} = \epsilon + \epsilon'$ .

We shall now prove this lemma by mathematical induction on the iteration number  $t$ . For  $t = 0$ , the invariant holds trivially as  $0 \leq \psi(\mathbf{C}^D[h^0]) \leq 1$ . Assume the invariant holds at the end of iteration  $t - 1 \in \{0, \dots, T - 1\}$ ; we shall prove that the invariant holds at the end of iteration  $t$ . In particular, we consider two cases at iteration  $t$ . In the first case,  $\psi(\hat{\Gamma}^t) \geq \gamma^t$ , leading to the assignments  $\alpha^t = \gamma^t$ ,  $\beta^t = \beta^{t-1}$ , and  $h^t = \hat{g}^t$ . We have from the definition of  $\epsilon'$

$$\begin{aligned} \langle \mathbf{A} - \gamma^t \mathbf{B}, \mathbf{C}^D[\hat{g}^t] \rangle &\geq \langle \mathbf{A} - \gamma^t \mathbf{B}, \hat{\Gamma}^t \rangle - \|\mathbf{A} - \gamma^t \mathbf{B}\|_1 \epsilon' \\ &= \langle \mathbf{A}, \hat{\Gamma}^t \rangle - \gamma^t \langle \mathbf{B}, \hat{\Gamma}^t \rangle - \|\mathbf{A} - \gamma^t \mathbf{B}\|_1 \epsilon' \\ &= \langle \mathbf{B}, \hat{\Gamma}^t \rangle (\psi(\hat{\Gamma}^t) - \gamma^t) - \|\mathbf{A} - \gamma^t \mathbf{B}\|_1 \epsilon' \\ &\geq 0 - \|\mathbf{A} - \gamma^t \mathbf{B}\|_1 \epsilon' \\ &> -\|\mathbf{A} - \gamma^t \mathbf{B}\|_1 (2\epsilon + \epsilon') \\ &\geq -(\|\mathbf{A}\|_1 + \|\mathbf{B}\|_1) (2\epsilon + \epsilon'), \end{aligned}$$

where the third step follows from our case assumption that  $\psi(\hat{\Gamma}^t) \geq \gamma^t$  and  $\langle \mathbf{B}, \hat{\Gamma}^t \rangle > 0$ , the fifth step follows from  $\epsilon > 0$ , and the last step follows from triangle inequality and  $\gamma^t \leq \sup_{\mathbf{C} \in \mathcal{C}_D} \psi(\mathbf{C}) \leq 1$ . The above inequality further gives us

$$\begin{aligned} \frac{\langle \mathbf{A}, \mathbf{C}^D[\hat{g}^t] \rangle}{\langle \mathbf{B}, \mathbf{C}^D[\hat{g}^t] \rangle} &> \gamma^t - \frac{\|\mathbf{A}\|_1 + \|\mathbf{B}\|_1}{\langle \mathbf{B}, \mathbf{C}^D[\hat{g}^t] \rangle} (2\epsilon + \epsilon') \\ &\geq \gamma^t - \frac{1}{b} (\|\mathbf{A}\|_1 + \|\mathbf{B}\|_1) (2\epsilon + \epsilon') \\ &= \gamma^t - \tau (2\epsilon + \epsilon') \\ &= \gamma^t - \tau \bar{\epsilon} \\ &= \alpha^t - \tau \bar{\epsilon}, \end{aligned}$$

where the second step follows from  $\min_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{B}, \mathbf{C} \rangle > b$  and the last step follows from the assignment  $\alpha^t = \gamma^t$ . In other words,

$$\psi(\mathbf{C}^D[h^t]) = \psi(\mathbf{C}^D[\hat{g}^t]) = \frac{\langle \mathbf{A}, \mathbf{C}^D[\hat{g}^t] \rangle}{\langle \mathbf{B}, \mathbf{C}^D[\hat{g}^t] \rangle} > \alpha^t - \tau\bar{\epsilon}.$$

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

$$\beta^t + \tau\bar{\epsilon} = \beta^{t-1} + \tau\bar{\epsilon} \geq \max_{\mathbf{C} \in \mathcal{C}_D} \psi(\mathbf{C}) \geq \psi(\mathbf{C}^D[h^t]) > \alpha^t - \tau\bar{\epsilon}.$$

Thus under the first case, the invariant holds at the end of iteration  $t$ .

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

$$\alpha^t - \tau\bar{\epsilon} = \alpha^{t-1} - \tau\bar{\epsilon} \leq \psi(\mathbf{C}^D[h^{t-1}]) = \psi(\mathbf{C}^D[h^t]). \quad (13)$$

Next, recall that  $\hat{\mathbf{L}}^t \in [0, 1]^{n \times n}$  is a scaled and translated version of  $-(\mathbf{A} - \gamma^t \mathbf{B})$ ; clearly, there exists  $c_t \in \mathbb{R}$  and  $0 < a_t \leq 2\|\mathbf{A} - \gamma^t \mathbf{B}\|_\infty$  such that  $\mathbf{A} - \gamma^t \mathbf{B} = c_t - a_t \hat{\mathbf{L}}^t$ . Then for  $\mathbf{C}^* = \operatorname{argmax}_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{A} - \gamma^t \mathbf{B}, \mathbf{C} \rangle$ , we have

$$\begin{aligned} \langle \mathbf{A} - \gamma^t \mathbf{B}, \mathbf{C}^* \rangle &= c_t - a_t \langle \hat{\mathbf{L}}^t, \mathbf{C}^* \rangle \\ &\leq c_t - a_t \langle \hat{\mathbf{L}}^t, \mathbf{C}^D[h^t] \rangle + a_t \epsilon \\ &\leq c_t - a_t \langle \hat{\mathbf{L}}^t, \mathbf{C}^D[h^t] \rangle + 2\|\mathbf{A} - \gamma^t \mathbf{B}\|_\infty \epsilon \\ &= \langle \mathbf{A} - \gamma^t \mathbf{B}, \mathbf{C}^D[h^t] \rangle + 2\|\mathbf{A} - \gamma^t \mathbf{B}\|_\infty \epsilon \\ &\leq \langle \mathbf{A} - \gamma^t \mathbf{B}, \hat{\Gamma}^t \rangle + \|\mathbf{A} - \gamma^t \mathbf{B}\|_1 \epsilon' + 2\|\mathbf{A} - \gamma^t \mathbf{B}\|_\infty \epsilon \\ &= \langle \mathbf{A}, \hat{\Gamma}^t \rangle - \gamma^t \langle \mathbf{B}, \hat{\Gamma}^t \rangle + \|\mathbf{A} - \gamma^t \mathbf{B}\|_1 \epsilon' + 2\|\mathbf{A} - \gamma^t \mathbf{B}\|_\infty \epsilon \\ &= \langle \mathbf{B}, \hat{\Gamma}^t \rangle (\psi(\hat{\Gamma}^t) - \gamma^t) + \|\mathbf{A} - \gamma^t \mathbf{B}\|_1 \epsilon' + 2\|\mathbf{A} - \gamma^t \mathbf{B}\|_\infty \epsilon \\ &\leq \langle \mathbf{B}, \hat{\Gamma}^t \rangle (0) + \|\mathbf{A} - \gamma^t \mathbf{B}\|_1 \epsilon' + 2\|\mathbf{A} - \gamma^t \mathbf{B}\|_\infty \epsilon \\ &\leq \|\mathbf{A} - \gamma^t \mathbf{B}\|_1 (2\epsilon + \epsilon') \\ &\leq (\|\mathbf{A}\|_1 + \|\mathbf{B}\|_1) (2\epsilon + \epsilon'), \end{aligned}$$

where the second step follows from Eq. (12), the third step uses  $a_t \leq \|\mathbf{A} - \gamma^t \mathbf{B}\|_\infty$ , the fifth step follows from the definition of  $\epsilon'$  and Holder's inequality, the seventh step follows from our case assumption that  $\psi(\hat{\Gamma}^t) \leq \gamma^t$  and  $\langle \mathbf{B}, \hat{\Gamma}^t \rangle > 0$ , and the last step follows from triangle inequality and  $\gamma^t \leq \sup_{\mathbf{C} \in \mathcal{C}_D} \psi(\mathbf{C}) \leq 1$ . In particular, we have for all  $\mathbf{C} \in \mathcal{C}_D$ ,

$$\langle \mathbf{A} - \gamma^t \mathbf{B}, \mathbf{C} \rangle \leq (\|\mathbf{A}\|_1 + \|\mathbf{B}\|_1) (2\epsilon + \epsilon'),$$

or

$$\frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle} \leq \gamma^t + \frac{\|\mathbf{A}\|_1 + \|\mathbf{B}\|_1}{\langle \mathbf{B}, \mathbf{C} \rangle} (2\epsilon + \epsilon').$$

Since  $\min_{\mathbf{C} \in \mathcal{C}_D} \langle \mathbf{B}, \mathbf{C} \rangle > b$ , we have from the above, for all  $\mathbf{C} \in \mathcal{C}_D$ ,

$$\psi(\mathbf{C}) \leq \gamma^t + \frac{1}{b} (\|\mathbf{A}\|_1 + \|\mathbf{B}\|_1) (2\epsilon + \epsilon') = \gamma^t + \tau\bar{\epsilon} = \beta^t + \tau\bar{\epsilon}.$$

In other words,

$$\sup_{\mathbf{C} \in \mathcal{C}_D} \psi(\mathbf{C}) \leq \beta^t + \tau\bar{\epsilon}.$$

By combining the above with Eq. (13), we can see that the invariant holds in iteration  $t$  under this case as well. This completes the proof of the lemma.  $\square$



**Lemma 28 (Multiplicative Progress in Each Iteration of Algorithm 2).** *Let  $\psi$  be as defined in Theorem 17. Then the following is true in each iteration  $1 \leq t \leq T$  of Algorithm 2:*

$$\beta^t - \alpha^t = \frac{1}{2}(\beta^{t-1} - \alpha^{t-1}).$$

*Proof.* We consider two cases in each iteration of Algorithm 2. If in an iteration  $t \in \{1, \dots, T\}$ ,  $\psi(\hat{\Gamma}^t) \geq \gamma^t$ , leading to the assignment  $\alpha^t = \gamma^t$ , then

$$\begin{aligned} \beta^t - \alpha^t &= \beta^{t-1} - \gamma^t \\ &= \beta^{t-1} - \frac{\alpha^{t-1} + \beta^{t-1}}{2} \\ &= \frac{1}{2}(\beta^{t-1} - \alpha^{t-1}). \end{aligned}$$

On the other hand, if  $\psi(\hat{\Gamma}^t) < \gamma^t$ , leading to the assignment  $\beta^t = \gamma^t$ , then

$$\begin{aligned} \beta^t - \alpha^t &= \gamma^t - \alpha^{t-1} \\ &= \frac{\alpha^{t-1} + \beta^{t-1}}{2} - \alpha^{t-1} \\ &= \frac{1}{2}(\beta^{t-1} - \alpha^{t-1}). \end{aligned}$$

Thus in both cases, the statement of the lemma is seen to hold.  $\square$

We now prove Theorem 17.

*Proof of Theorem 17.* For the classifier  $h_S^{\text{BS}} = h^T$  output by Algorithm 2 after  $T$  iterations, we have from Lemma 27

$$\begin{aligned} \mathcal{P}_D^* - \mathcal{P}_D[h_S^{\text{BS}}] &= \sup_{\mathbf{C} \in \mathcal{C}_D} \psi(\mathbf{C}) - \psi(\mathbf{C}^D[h^T]) \\ &< (\beta^t + \tau\bar{\epsilon}) - (\alpha^t - \tau\bar{\epsilon}) \\ &= \beta^t - \alpha^t + 2\tau\bar{\epsilon} \\ &\leq 2^{-T}(\beta^0 - \alpha^0) + 2\tau\bar{\epsilon} \\ &= 2^{-T}(1 - 0) + 2\tau\bar{\epsilon} \\ &= 2^{-T} + 2\tau\bar{\epsilon}, \end{aligned}$$

where  $\bar{\epsilon}$  is as defined in Lemma 27; the fifth step above follows from Lemma 28. Setting  $T = \kappa m$  thus gives us

$$\mathcal{P}_D^* - \mathcal{P}_D[h_S^{\text{BS}}] \leq 2\tau\mathbf{E}_X[\|\hat{\boldsymbol{\eta}}(X) - \boldsymbol{\eta}(X)\|_1] + 2\tau \sup_{h \in \mathcal{H}_\eta} \|\mathbf{C}^D[h] - \hat{\mathbf{C}}^{S_2}[h]\|_\infty + 2^{-\kappa m}.$$

By an application Lemma 15 to the second term in the right-hand side of the above inequality (noting that  $|S_2| = \lfloor m/2 \rfloor$ ), we then have for any  $\delta > 0$ , with probability at least  $1 - \delta$ ,

$$\mathcal{P}_D^* - \mathcal{P}_D[h_S^{\text{BS}}] \leq 2\tau\mathbf{E}_X[\|\hat{\boldsymbol{\eta}}(X) - \boldsymbol{\eta}(X)\|_1] + 2\sqrt{2}C\tau\sqrt{\frac{n^2 \log(n) \log(m) + \log(n^2/\delta)}{m}} + 2^{-\kappa m},$$

for a distribution-independent constant  $C > 0$ .  $\square$

### C.5. Extending Algorithm 1 to Non-Smooth Performance Measures

In Section 5, we showed that Algorithm 1 was consistent for any concave smooth performance measure (see Theorem 16). We now extend this result to concave performance measures for which the associated  $\psi$  is non-smooth (but differentiable); these include the G-mean, H-mean and Q-mean performance measures in 1. In particular, for these performance measures, we prescribe that Algorithm 1 be applied to a suitable smooth approximation to  $\psi$ ; if the quality of this approximation improves with the size of the given training sample (at an appropriate rate), then the resulting algorithm can be shown to be consistent for the original performance measure.

**Theorem 29.** Let  $\psi : [0, 1]^{n \times n} \rightarrow \mathbb{R}_+$  be such that for any  $\rho \in \mathbb{R}_+$ , there exists  $\psi_\rho : [0, 1]^{n \times n} \rightarrow \mathbb{R}_+$  which is concave over  $\mathcal{C}_D$ ,  $L_\rho$  Lipschitz and  $\beta_\rho$  smooth w.r.t. the  $\ell_1$  norm with

$$\sup_{\mathbf{C} \in \mathcal{C}_D} |\psi(\mathbf{C}) - \psi_\rho(\mathbf{C})| \leq \theta(\rho),$$

for some strictly increasing function  $\theta : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ . Let  $S = (S_1, S_2) \in (\mathcal{X} \times [n])^m$  be a training sample drawn i.i.d. from  $D$ . Further, let  $\hat{\eta} : \mathcal{X} \rightarrow \Delta_n$  be the CPE model learned from  $S_1$  in Algorithm 1 and  $h_S^{\text{FW}, \rho} : \mathcal{X} \rightarrow \Delta_n$  be the classifier obtained after  $\kappa m$  iterations by Algorithm 1 when run for the performance measure  $\psi_\rho$ . Then for any  $\delta \in (0, 1]$ , with probability at least  $1 - \delta$  (over draw of  $S$  from  $D^m$ ),

$$\mathcal{P}_D^{\psi, *} - \mathcal{P}_D^\psi[h_S^{\text{FW}, \rho}] \leq 4L_\rho \mathbf{E}_X [\|\hat{\eta}(X) - \eta(X)\|_1] + 4\sqrt{2}\beta_\rho n^2 C \sqrt{\frac{n^2 \log(n) \log(m) + \log(n^2/\delta)}{m}} + \frac{8\beta_\rho}{\kappa m + 2} + 2\theta(\rho),$$

where  $C > 0$  is a distribution-independent constant.

*Proof.* From Theorem 16 we have that

$$\mathcal{P}_D^{\psi_\rho, *} - \mathcal{P}_D^{\psi_\rho}[h_S^{\text{FW}, \rho}] \leq 4L_\rho \mathbf{E}_X [\|\hat{\eta}(X) - \eta(X)\|_1] + 4\sqrt{2}\beta_\rho n^2 C \sqrt{\frac{n^2 \log(n) \log(m) + \log(n^2/\delta)}{m}} + \frac{8\beta_\rho}{\kappa m + 2}. \quad (14)$$

For simplicity assume that the  $\psi$ -optimal classifier exists; the proof can be easily extended when this is not the case. Let  $h^* : \mathcal{X} \rightarrow \Delta_n$  be a  $\psi$ -optimal classifier; note that this classifier need not be  $\psi_\rho$ -optimal. We then have that

$$\begin{aligned} \mathcal{P}_D^{\psi, *} - \mathcal{P}_D^\psi[h_S^{\text{FW}, \rho}] &= \psi(\mathbf{C}^D[h^*]) - \psi(\mathbf{C}^D[h_S^{\text{FW}, \rho}]) \\ &\leq \psi_\rho(\mathbf{C}^D[h^*]) - \psi_\rho(\mathbf{C}^D[h_S^{\text{FW}, \rho}]) + 2\theta(\rho) \\ &= \mathcal{P}_D^{\psi_\rho}[h^*] - \mathcal{P}_D^{\psi_\rho}[h_S^{\text{FW}, \rho}] + 2\theta(\rho) \\ &\leq \mathcal{P}_D^{\psi_\rho, *} - \mathcal{P}_D^{\psi_\rho}[h_S^{\text{FW}, \rho}] + 2\theta(\rho) \\ &\leq 4L_\rho \mathbf{E}_X [\|\hat{\eta}(X) - \eta(X)\|_1] + 4\sqrt{2}\beta_\rho n^2 C \sqrt{\frac{n^2 \log(n) \log(m) + \log(n^2/\delta)}{m}} + \frac{8\beta_\rho}{\kappa m + 2} + 2\theta(\rho), \end{aligned}$$

where the second step follows from our assumption that  $\sup_{\mathbf{C} \in \mathcal{C}_D} |\psi(\mathbf{C}) - \psi_\rho(\mathbf{C})| \leq \theta(\rho)$ , and the fifth step follows from the definition of  $\mathcal{P}_D^{\psi_\rho, *}$  and the last step uses Eq. (14). This completes the proof.  $\square$

We note that for each of G-mean, H-mean and Q-mean, one can construct a Lipschitz smooth approximation  $\psi_\rho$  as required in the above theorem. Now, suppose the CPE algorithm in Algorithm 1 is such that the class probability estimation error term in the theorem  $\mathbf{E}_X [\|\hat{\eta}(X) - \eta(X)\|_1] \xrightarrow{P} 0$  (as the number of training examples  $m \rightarrow \infty$ ). Then for each of the given performance measures, one can allow the parameter  $\rho$  (that determines the approximation quality of  $\psi_\rho$ ) to go to 0 as  $m \rightarrow \infty$  (at appropriate rate), so that the right-hand side of the bound in the theorem goes to 0 (as  $m \rightarrow \infty$ ), implying that Algorithm 1 is  $\psi$ -consistent. We postpone the details to a longer version of the paper.

## D. Supplementary Material for Section 6 (Experiments)

### D.1. Computation of Class Probability Function for Distribution Considered in Synthetic Data Experiments

We provide the calculations for the class probability function for the distribution considered in synthetic data experiments in Section 6. We present this for a more general distribution over  $\mathbb{R}^d \times [n]$ , where for each class  $i$ , the class prior probability is  $\pi_i$  and the class conditional distribution is a Gaussian distribution with mean  $\mu_i \in \mathbb{R}^d$  and the same (symmetric positive semidefinite) covariance matrix  $\Sigma \in \mathbb{R}^{d \times d}$ . We shall denote the pdf for the Gaussian corresponding to class  $i$  as  $f_i(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x - \mu_i)^\top \Sigma^{-1}(x - \mu_i)\right)$ . The class probability function for this distribution is then given by

$$\begin{aligned} \eta_i(x) &= \mathbf{P}(Y = i | X = x) \\ &= \frac{\pi_i f_i(x)}{\sum_{j=1}^n \pi_j f_j(x)} \end{aligned}$$

Table 6. Data sets used in experiments in Sections 6.2–6.4.

	Data set	# instances	# features	# classes
UCI	car	1728	21	4
	pageblocks	5473	10	5
	glass	214	9	6
	satimage	6435	36	6
	covtype	581012	54	7
	yeast	1484	8	10
	abalone	4177	10	12
IR	cora	2708	1433	4
	news20	12199	61188	4
	rcv1	15564	47236	11

$$\begin{aligned}
 &= \frac{\exp\left(-\frac{1}{2}(x - \mu_i)^\top \Sigma^{-1}(x - \mu_i) + \ln \pi_i\right)}{\sum_{j=1}^n \exp\left(-\frac{1}{2}(x - \mu_j)^\top \Sigma^{-1}(x - \mu_j) + \ln \pi_j\right)} \\
 &= \frac{\exp\left(\mu_i^\top \Sigma^{-1}x - \frac{1}{2}\mu_i^\top \Sigma^{-1}\mu_i + \ln \pi_i\right) \exp\left(-\frac{1}{2}x^\top \Sigma^{-1}x\right)}{\sum_{j=1}^n \exp\left(\mu_j^\top \Sigma^{-1}x - \frac{1}{2}\mu_j^\top \Sigma^{-1}\mu_j + \ln \pi_j\right) \exp\left(-\frac{1}{2}x^\top \Sigma^{-1}x\right)} \\
 &= \frac{\exp(w_i^\top x + b_i)}{\sum_{j=1}^n \exp(w_j^\top x + b_j)},
 \end{aligned}$$

where  $w_i = \Sigma^{-1}\mu_i$  and  $b_i = -\frac{1}{2}\mu_i^\top \Sigma^{-1}\mu_i + \ln \pi_i$ . Clearly, the class probability function for the distribution considered can be obtained as a softmax of a linear function.

## D.2. Additional Experimental Details/Results

In all our experiments, the regularization parameter for each algorithm was chosen from the range  $\{10^{-4}, \dots, 10^4\}$  using a held-out portion of the training set.

**Synthetic data experiments.** Since the distribution used to generate synthetic data and the four performance measures considered satisfy the condition in Theorem 13, the optimal classifier for each performance measure can be obtained by computing the  $\psi^{\mathbf{L}^*}$ -optimal classifier for some loss matrix  $\mathbf{L}^* \in [0, 1]^{n \times n}$ ; we have a similar characterization for micro  $F_1$  using Theorem 13. In our experiments, we computed the optimal performance for a given performance measure by performing a search over a large range of  $n \times n$  loss matrices  $\mathbf{L}$ , used the true conditional class probability to compute a  $\psi^{\mathbf{L}}$ -optimal classifier for each such  $\mathbf{L}$  (see Proposition 6), and chose among these classifiers the one which gave the highest performance value (on a large sample drawn from the distribution). Moreover, since the class probability function here is a softmax of linear functions, it follows that the Bayes optimal performance is also achieved by a linear classification model, and therefore learning a linear model suffices to achieve consistency; we therefore learn a linear classification model in all experiments. Also, recall that Algorithm 1 outputs a randomized classifier, while Algorithm 2 outputs a deterministic classifier. In our experimental results, the  $\psi$ -performance of a randomized classifier was evaluated using the ‘expected’ (empirical) confusion matrix of the deterministic classifiers in its support.

**Real data experiments.** All real data sets used in our experiments have been listed in Table 6. The version of the CoRA data set used in our experiments was obtained from <http://membres-lig.imag.fr/grimal/data.html>. The 20 Newsgroup data was obtained from <http://qwone.com/~jason/20Newsgroups/>. For the RCV1 data, we used a preprocessed version obtained from <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>. For each of the UCI data sets used in our experiments, the training set was normalized to 0 mean and unit variance, and this transformation was applied on the test set. Table 7–9 contains results on UCI data sets not provided in Section 6. Table 10–12 contains training times for Algorithm 1 (applied to the G-mean, H-mean and Q-mean measures) and the baseline SVM<sup>perf</sup> and 0-1 logistic regression methods on all UCI data sets; in each case, the symbol  $\times$  against SVM<sup>perf</sup> indicates the method did not complete after 96 hrs.

**Implementation details.** The proposed Frank-Wolfe based and bisection based algorithms were implemented in MATLAB; in order to learn a CPE model in these algorithms, we used the multiclass logistic regression solver provided in <http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html> for the experiments on the synthetic and UCI data sets, and the liblinear logistic regression implementation provided in [www.csie.ntu.edu.tw/~cjlin/](http://www.csie.ntu.edu.tw/~cjlin/)

	car	pageblocks	glass	satimage	covtype	yeast	abalone
Frank-Wolfe (GM)	<b>0.945</b>	<b>0.908</b>	<b>0.680</b>	<b>0.843</b>	<b>0.695</b>	<b>0.448</b>	<b>0.223</b>
SVM <sup>perf</sup> (GM)	0.792	0.796	0.431	×	×	×	×
LogReg (0-1)	0.911	0.691	0.146	0.779	0.692	0.000	0.000

Table 7. Results for G-mean on UCI data sets.

	car	pageblocks	glass	satimage	covtype	yeast	abalone
Frank-Wolfe (HM)	<b>0.945</b>	<b>0.904</b>	<b>0.632</b>	<b>0.836</b>	<b>0.686</b>	<b>0.412</b>	<b>0.197</b>
SVM <sup>perf</sup> (HM)	0.880	0.574	0.381	×	×	×	×
LogReg (0-1)	0.909	0.631	0.143	0.731	0.679	0.000	0.000

Table 8. Results for H-mean on UCI data sets.

	car	pageblocks	glass	satimage	covtype	yeast	abalone
Frank-Wolfe (QM)	<b>0.930</b>	<b>0.877</b>	<b>0.613</b>	<b>0.821</b>	<b>0.685</b>	<b>0.510</b>	<b>0.247</b>
SVM <sup>perf</sup> (QM)	0.909	0.651	0.481	×	×	×	×
LogReg (0-1)	0.898	0.660	0.490	0.725	0.675	0.473	0.223

Table 9. Results for Q-mean on UCI data sets.

	car	pageblocks	glass	satimage	covtype	yeast	abalone
Frank-Wolfe (GM)	1.96	5.89	0.27	9.66	139.60	1.68	7.31
SVM <sup>perf</sup> (GM)	8327.54	63667.67	1302.84	×	×	×	×
LogReg (0-1)	0.59	1.70	0.07	4.48	106.27	0.40	3.84

Table 10. Training time (in secs) for G-mean on UCI data sets.

	car	pageblocks	glass	satimage	covtype	yeast	abalone
Frank-Wolfe (HM)	1.96	5.85	0.26	9.02	125.30	1.69	7.14
SVM <sup>perf</sup> (HM)	3342.08	35836.87	108.80	×	×	×	×
LogReg (0-1)	0.57	1.55	0.07	4.78	127.12	0.38	4.07

Table 11. Training time (in secs) for H-mean on UCI data sets.

	car	pageblocks	glass	satimage	covtype	yeast	abalone
Frank-Wolfe (QM)	1.93	6.11	0.27	9.00	134.85	1.65	7.29
SVM <sup>perf</sup> (QM)	6795.87	54803.42	158.48	×	×	×	×
LogReg (0-1)	0.61	1.79	0.07	4.72	120.60	0.43	3.84

Table 12. Training time (in secs) for Q-mean on UCI data sets.

`liblinear` for the experiments on IR data. All run-time experiments were run on Intel Xeon quad-core machines (2.66 GHz, 12 MB cache) with 16 GB RAM.

We implemented SVM<sup>perf</sup> using a publicly available structural SVM API<sup>9</sup>. The SVM<sup>perf</sup> method (proposed originally for binary performance measures (Joachims, 2005)) uses a cutting plane solver where computing the most-violated constraint requires a search over all valid confusion matrices for the given training sample. In the case of the G-mean, H-mean and Q-mean measures, this search can be restricted to the diagonal entries of the confusion matrix, but will still require (in the worst case) time exponential in the number of classes; in the case of the micro  $F_1$ , this search is more expensive and will involve searching over  $3n - 3$  entries of the confusion matrix. While we use an exact implementation of SVM<sup>perf</sup> for these three performance measures, for the micro  $F_1$ -measure, we use a version that optimizes an approximation to the micro  $F_1$  (in particular, optimizes the variant of micro  $F_1$  analyzed by (Parambath et al., 2014)) and requires fewer computations. The tolerance parameter for the cutting-plane method in SVM<sup>perf</sup> was set to 0.01 for all experiments except on the Pageblocks and CoRA data sets, where the tolerance was set to 0.1 to enable faster run-time.

<sup>9</sup>[http://www.cs.cornell.edu/people/tj/svm\\_light/svm\\_struct.html](http://www.cs.cornell.edu/people/tj/svm_light/svm_struct.html)