Consistent Multiclass Algorithms for Complex Performance Measures

Harikrishna Narasimhan[*] HARIKRISHNA@CSA.IISC.ERNET.IN
Harish G. Ramaswamy[*] HARISH_GURUP@CSA.IISC.ERNET.IN
Aadirupa Saha AADIRUPA.SAHA@CSA.IISC.ERNET.IN
Shivani Agarwal SHIVANI@CSA.IISC.ERNET.IN
Indian Institute of Science, Bangalore 560012, INDIA

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$^{\text{perf}}$ 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^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

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$^{\text{perf}}$ both in terms of generalization performance and in terms of training time.

Notation. For $n \in \mathbb{Z}_+$, we denote $[n] = \{1, \ldots, n\}$ 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 $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (\mathcal{X} \times [n])^m$, the goal is to learn a multiclass classifier $h_S : \mathcal{X} \to [n]$, or more generally, a *randomized* multiclass classifier $h_S : \mathcal{X} \to \Delta_n$ (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 μ 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*

$$C^D_{ij}[h] = \mathbf{P}\big(Y = i, h(X) = j\big)\,,$$

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, $\sum_{i,j} C^D_{ij}[h] = 1$.

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

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

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 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 $\mathbf{C}^D[h]$:*

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

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

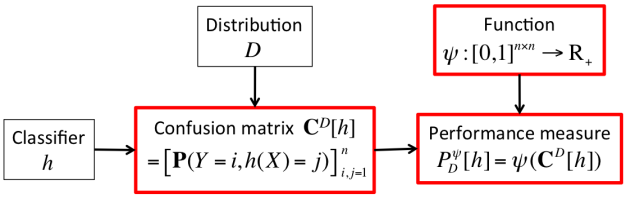

Figure 1. Complex multiclass performance measures, given by arbitrary functions of the confusion matrix, generalize both common loss-based performance measures, and binary performance measures expressed in terms of TP, TN, FP and FN. (In practice, the distribution D is unknown; one estimates the confusion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

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 $\psi^{\text{AM}}(\mathbf{C}) = \frac{1}{2}\big(\frac{\text{TP}}{\text{TP+FN}} + \frac{\text{TN}}{\text{TN+FP}}\big)$, the F_β-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).*[1]

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*

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

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

Table 1. Examples of complex multiclass performance measures.

Performance measure	$\psi(\mathbf{C})$
G-mean	$\left(\prod_{i=1}^{n} \frac{C_{ii}}{\sum_{j=1}^{n} C_{ij}}\right)^{1/n}$
H-mean	$n\left(\sum_{i=1}^{n} \frac{\sum_{j=1}^{n} C_{ij}}{C_{ii}}\right)^{-1}$
Q-mean	$1 - \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{C_{ii}}{\sum_{j=1}^{n} C_{ij}}\right)^2}$
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*[2]

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

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 ψ, in that they converge (in probability) to the optimal ψ-performance as the training sample size increases:

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

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

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

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

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

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

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

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

As a first step towards designing ψ-consistent algorithms, we start by examining the form of ψ-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 $\psi^{\mathbf{L}} : [0,1]^{n\times n}\to\mathbb{R}_+$ 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*

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

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 $\psi : [0,1]^{2\times 2}\to\mathbb{R}_+$ 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 ψ-optimal classifier of one of the following forms: $h^*(x) = \text{sign}(\eta_1(x) - \theta_D^*)$ or $h^*(x) = \text{sign}(\theta_D^* - \eta_1(x))$, where $\theta_D^* \in [0,1]$ depends on a_{ij}'s and b_{ij}'s, and on the optimal ψ-performance $\mathcal{P}_D^{\psi,*}$.*[3,4]

[3]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.

[4]The original result of Koyejo et al. (2014) makes a continuity assumption on the marginal distribution μ; 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 $\psi : [0,1]^{2\times 2} \to \mathbb{R}_+$ 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 ψ-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 ψ, we will find it useful to view the optimal ψ-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:*

$$\mathcal{C}_D = \left\{ \mathbf{C}^D[h] : \; h : \mathcal{X} \to \Delta_n \right\}.$$

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

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

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 ψ, and for 'monotonic' performance measures ψ under a mild continuity assumption on D, an optimal classifier does indeed exist, and moreover, in each case, a ψ-optimal classifier can be obtained by finding a certain loss-based optimal classifier.

Theorem 11 (Form of Bayes optimal classifier for ratio-of–linear ψ). *Let $\psi : [0,1]^{n\times n} \to \mathbb{R}_+$ 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 $t^*_D = \mathcal{P}^{\psi,*}_D$. Let $\widetilde{\mathbf{L}}^* = -(\mathbf{A} - t^*_D \mathbf{B})$, and let $\mathbf{L}^* \in [0,1]^{n\times n}$ be obtained by scaling and shifting $\widetilde{\mathbf{L}}^*$ so its entries lie in $[0,1]$. Then any classifier that is $\psi^{\mathbf{L}^*}$-optimal is also ψ-optimal.*

Lemma 12 (Existence of Bayes optimal classifier for monotonic ψ). *Let D be such that the probability measure associated with the random vector $\boldsymbol{\eta}(X) = (\eta_1(X), \ldots, \eta_n(X))^\top$ is absolutely continuous w.r.t. the base probability measure associated with the uniform distribution over Δ_n, and let ψ 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 $\exists h^* : \mathcal{X} \to \Delta_n$ s.t. $\mathcal{P}^{\psi}_D[h^*] = \mathcal{P}^{\psi,*}_D$.*

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

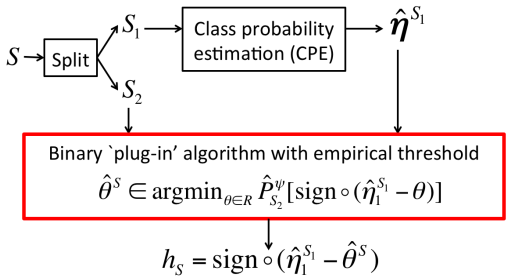

Figure 2. 'Plug-in' algorithm used for binary performance measures ψ (Koyejo et al., 2014; Narasimhan et al., 2014). In practice, one searches over $O(|S_2|)$ values of the threshold θ. In the multiclass case, such a method requires searching over an exponential number of loss matrices and is computationally 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 ψ-optimal performance.[5] 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 $\mathcal{P}^{\psi,*}_D$ in the ratio-of-linear case, or a ψ-optimal classifier h^* or ψ-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 ψ-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.[6]

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.

[5] 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.

[6] 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.

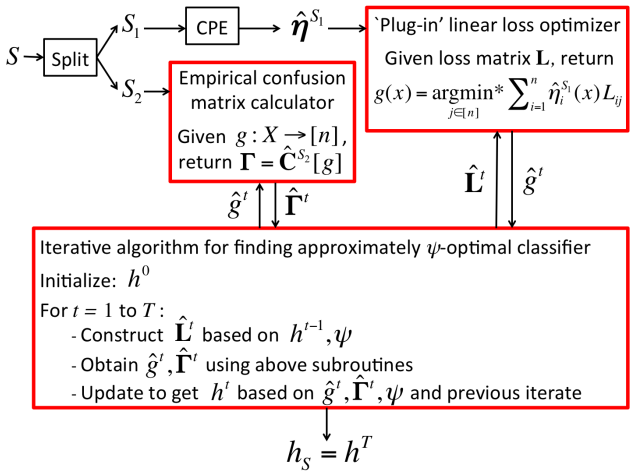

Figure 3. Overall framework of the multiclass learning algorithms proposed in this paper. The algorithms solve (approximately) $\max_{\mathbf{C}\in\mathcal{C}_D} \psi(\mathbf{C})$, by using an optimization method that on each iteration requires only solving a linear loss minimization problem and calculating 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 ψ-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 $\boldsymbol{\eta} : \mathcal{X} \to \Delta_n$ – or an estimate $\widehat{\boldsymbol{\eta}} : \mathcal{X} \to \Delta_n$ – one operation that is easy to perform is to find an optimal classifier for a *linear* loss $\mathbf{L}$: one simply returns the classifier $g : \mathcal{X} \to [n]$ given by $g(x) = \operatorname{argmin}^*_{j\in[n]} \sum_{i=1}^n \widehat{\eta}_i(x) L_{ij}$. 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 ψ, 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 ψ 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 γ by again appealing to a linear minimization subroutine, leading to an efficient binary search type algorithm based on the bisection method.

Algorithm 1 Algorithm Based on Frank-Wolfe Method

Input: $\psi : [0,1]^{n\times n} \to \mathbb{R}_+$
$S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (\mathcal{X} \times [n])^m$
Parameter: $\kappa \in \mathbb{N}$
Split S into S_1 and S_2 with sizes $\lceil \frac{m}{2} \rceil$ and $\lfloor \frac{m}{2} \rfloor$
$\widehat{\boldsymbol{\eta}} = \text{CPE}(S_1)$
Initialize: $h^0 : \mathcal{X} \to \Delta_n$, $\widehat{\mathbf{C}}^0 = \widehat{\mathbf{C}}^{S_2}[h^0]$
For $t = 1$ **to** $T = \kappa m$ **do**
  $\widehat{\mathbf{L}}^t = -\nabla\psi(\widehat{\mathbf{C}}^{t-1})$, scaled and shifted to $[0,1]^{n\times n}$
  Obtain $\widehat{g}^t : x \mapsto \operatorname{argmin}^*_{j\in[n]} \sum_{i=1}^n \widehat{\eta}_i(x) \widehat{L}^t_{ij}$
  $\widehat{\boldsymbol{\Gamma}}^t = \widehat{\mathbf{C}}^{S_2}[\widehat{g}^t]$
  $h^t = \left(1 - \frac{2}{t+1}\right) h^{t-1} + \frac{2}{t+1} \widehat{g}^t$
  $\widehat{\mathbf{C}}^t = \left(1 - \frac{2}{t+1}\right) \widehat{\mathbf{C}}^{t-1} + \frac{2}{t+1} \widehat{\boldsymbol{\Gamma}}^t$
end For
Output: $h_S^{\text{FW}} = h^T : \mathcal{X} \to \Delta_n$

Both algorithms divide the input training sample S into a part S_1 used for obtaining a class probability estimate $\widehat{\boldsymbol{\eta}}^{S_1}$, 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 ψ 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 $\widehat{\boldsymbol{\eta}}^{S_1}$, 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 ψ 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 ψ over an empirical constraint set, but instead maximizes ψ

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 ψ is greater than γ, 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 ψ 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 α^t and β^t on the maximal value of ψ, determine whether this maximal value is greater than the midpoint γ^t of these bounds using the linear minimization subroutine, and then update $\mathbf{C}^t$ and α^t, β^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.[7]

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 ψ-consistent. Our proofs rely on convergence guarantees of the underlying optimization methods, together with Lemmas 14 and 15 below, which yield approximation guarantees for the plug-in linear loss minimization and empirical confusion matrix calculation steps, respectively.

[7] 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

Input: $\psi(\mathbf{C}) = \frac{\langle \mathbf{A}, \mathbf{C} \rangle}{\langle \mathbf{B}, \mathbf{C} \rangle}$ with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$
$S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (\mathcal{X} \times [n])^m$
Parameter: $\kappa \in \mathbb{N}$
Split S into S_1 and S_2 with sizes $\lceil \frac{m}{2} \rceil$ and $\lfloor \frac{m}{2} \rfloor$
$\widehat{\boldsymbol{\eta}} = \text{CPE}(S_1)$
Initialize: $h^0 : \mathcal{X} \to [n]$, $\alpha^0 = 0$, $\beta^0 = 1$
For $t = 1$ to $T = \kappa m$ **do**
  $\gamma^t = (\alpha^{t-1} + \beta^{t-1})/2$
  $\widehat{\mathbf{L}}^t = -(\mathbf{A} - \gamma^t \mathbf{B})$, scaled and shifted to $[0, 1]^{n \times n}$
  Obtain $\widehat{g}^t : x \mapsto \operatorname{argmin}^*_{j \in [n]} \sum_{i=1}^n \widehat{\eta}_i(x) \widehat{L}^t_{ij}$
  $\widehat{\mathbf{\Gamma}}^t = \widehat{\mathbf{C}}^{S_2}[\widehat{g}^t]$
  If $\psi(\widehat{\mathbf{\Gamma}}^t) \geq \gamma^t$ **then** $\alpha^t = \gamma^t$, $\beta^t = \beta^{t-1}$, $h^t = \widehat{g}^t$
  else $\alpha^t = \alpha^{t-1}$, $\beta^t = \gamma^t$, $h^t = h^{t-1}$
end For
Output: $h^{\text{BS}}_S = h^T : \mathcal{X} \to [n]$

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

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

Lemma 15 (Uniform convergence of confusion matrices)**.** *Let $\mathbf{q} : \mathcal{X} \to \Delta_n$ and let $\mathcal{H}_{\mathbf{q}}$ be the set of (deterministic) classifiers $h : \mathcal{X} \to [n]$ that satisfy $h(x) = \operatorname{argmin}^*_{j \in [n]} \sum_{i=1}^n q_i(x) L_{ij}$ 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 $S \sim D^m$),*

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

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

5.1. Consistency of Frank-Wolfe Based Algorithm

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

Theorem 16 (ψ-regret of Frank-Wolfe based algorithm)**.** *Let $\psi : [0, 1]^{n \times n} \to \mathbb{R}_+$ be concave over $\mathcal{C}_D$, and L-Lipschitz and β-smooth w.r.t. the ℓ_1 norm. Let $S \in (\mathcal{X} \times [n])^m$ be drawn randomly from D^m. Let $\widehat{\boldsymbol{\eta}} : \mathcal{X} \to \Delta_n$ be the* CPE *model learned in Algorithm 1 and $h^{\text{FW}}_S : \mathcal{X} \to \Delta_n$ the classifier returned after κm iterations. Let $\delta \in (0, 1]$. Then with probability $\geq 1 - \delta$ (over $S \sim D^m$),*

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

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 $\widehat{\boldsymbol{\eta}}$ is learned by a CPE algorithm that guarantees $\mathbf{E}_X[\|\widehat{\boldsymbol{\eta}}(X) - \boldsymbol{\eta}(X)\|_1] \to 0$ as $m \to \infty$, as is done by any algorithm that minimizes a strictly proper composite multiclass loss over a suitably large function class (Vernet et al., 2011), then the above result yields ψ-consistency of the Frank-Wolfe based algorithm. For concave non-smooth performance measures ψ such as the G-mean, H-mean and Q-mean, Algorithm 1 can be applied to a suitable smooth approximation to ψ; 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 ψ-regret of Algorithm 2 for ratio-of-linear performance measures ψ:

Theorem 17 (ψ-regret of bisection based algorithm)**.** *Let $\psi : [0,1]^{n \times n} \to \mathbb{R}_+$ 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 $\widehat{\boldsymbol{\eta}} : \mathcal{X} \to \Delta_n$ be the* CPE *model learned in Algorithm 2 and $h_S^{\text{BS}} : \mathcal{X} \to [n]$ the classifier returned after κm iterations. Let $\delta \in (0,1]$. Then with probability $\geq 1 - \delta$ (over $S \sim D^m$),*

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

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 $\mathbf{E}_X[\|\widehat{\boldsymbol{\eta}}(X) - \boldsymbol{\eta}(X)\|_1] \to 0$ as $m \to \infty$, then the above result yields ψ-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$^{\text{perf}}$ algorithm (Joachims, 2005) and a standard multiclass logistic regression algorithm that optimizes 0-1 accuracy.[8] We note that the worst-case running time of SVM$^{\text{perf}}$ is exponential in the number of classes, and hence this method could not be scaled to data sets with large numbers of classes.

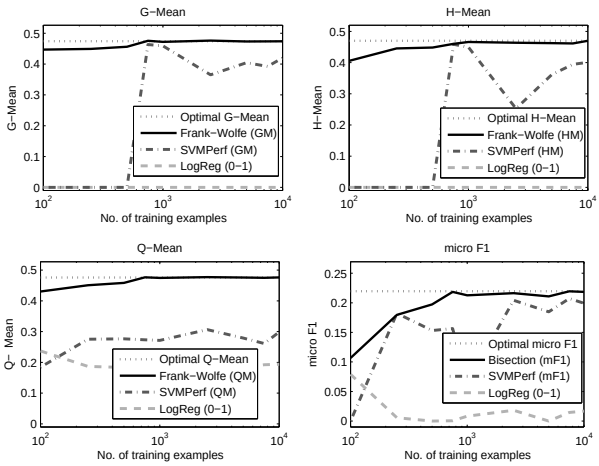

Figure 4. Convergence to Bayes optimal performance for G-mean, H-mean, Q-mean and micro F_1 measures on synthetic data.

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 $\left[\begin{smallmatrix} 5 & 1 \\ 1 & 5 \end{smallmatrix}\right]$. The class probability function $\boldsymbol{\eta} : \mathbb{R}^2 \to \Delta_3$ 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$^{\text{perf}}$ also seems to approach the optimal performance, but at a slower rate.

[8] 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	**0.945**	**0.908**	**0.680**	**0.223**
	SVM$^{\text{perf}}$	0.792	0.796	0.431	$\times$
	LogReg (0-1)	0.911	0.691	0.146	0.000
H-mean	Frank-Wolfe	**0.945**	**0.904**	**0.632**	**0.197**
	SVM$^{\text{perf}}$	0.880	0.574	0.381	$\times$
	LogReg (0-1)	0.909	0.631	0.143	0.000
Q-mean	Frank-Wolfe	**0.930**	**0.877**	**0.613**	**0.247**
	SVM$^{\text{perf}}$	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$^{\text{perf}}$ 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$^{\text{perf}}$ 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 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).

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	**0.690**	**0.772**	**0.502**
	SVM$^{\text{perf}}$	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$^{\text{perf}}$	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$^{\text{perf}}$	18095.98	$\times$	$\times$	
	LogReg (0-1)	0.08	19.04	11.88	

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$^{\text{perf}}$ 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$^{\text{perf}}$, 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., Kotlowski, W., Waegeman, W., and Hullermeier, 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.