

---

# Convex Calibrated Surrogates for the Multi-Label F-Measure

---

Mingyuan Zhang<sup>1</sup> Harish G. Ramaswamy<sup>2</sup> Shivani Agarwal<sup>1</sup>

## Abstract

The  $F$ -measure is a widely used performance measure for multi-label classification, where multiple labels can be active in an instance simultaneously (e.g. in image tagging, multiple tags can be active in any image). In particular, the  $F$ -measure explicitly balances recall (fraction of active labels predicted to be active) and precision (fraction of labels predicted to be active that are actually so), both of which are important in evaluating the overall performance of a multi-label classifier. As with most discrete prediction problems, however, directly optimizing the  $F$ -measure is computationally hard. In this paper, we explore the question of designing convex surrogate losses that are *calibrated* for the  $F$ -measure – specifically, that have the property that minimizing the surrogate loss yields (in the limit of sufficient data) a Bayes optimal multi-label classifier for the  $F$ -measure. We show that the  $F$ -measure for an  $s$ -label problem, when viewed as a  $2^s \times 2^s$  loss matrix, has rank at most  $s^2 + 1$ , and apply a result of Ramaswamy et al. (2014) to design a family of convex calibrated surrogates for the  $F$ -measure. The resulting surrogate risk minimization algorithms can be viewed as decomposing the multi-label  $F$ -measure learning problem into  $s^2 + 1$  binary class probability estimation problems. We also provide a quantitative regret transfer bound for our surrogates, which allows any regret guarantees for the binary problems to be transferred to regret guarantees for the overall  $F$ -measure problem, and discuss a connection with the algorithm of Dembczynski et al. (2013). Our experiments confirm our theoretical findings.

## 1. Introduction

The  $F_\beta$ -measure is a widely used performance measure for multi-label classification (MLC) problems. In particular, in an MLC problem, multiple labels can be active in an instance simultaneously; a good example is that of image tagging, where several tags (such as *sky*, *sand*, *water*) can be active in the same image. In such problems, when evaluating the performance of a classifier on a particular instance, it is important to balance the *recall* of the classifier on the given instance, i.e. the fraction of active labels for that instance that are correctly predicted as such, and the *precision* of the classifier on the instance, i.e. the fraction of labels predicted to be active for that instance that are actually so. The  $F_\beta$ -measure accomplishes this by taking the (possibly weighted) harmonic mean of these two quantities.

Unfortunately, as with most discrete prediction problems, optimizing the  $F_\beta$ -measure directly during training is computationally hard. Consequently, one generally settles for some form of approximation. One approach is to simply treat the labels as independent, and train a separate binary classifier for each label; this is sometimes referred to as the *binary relevance* (BR) approach. Of course, this ignores the fact that labels can have correlations among them (e.g. *sky* and *cloud* may be more likely to co-occur than *sky* and *computer*). Several other approaches have been proposed in recent years (Dembczynski et al., 2013; Koyejo et al., 2015; Wu & Zhou, 2017; Pillai et al., 2017).

In this paper, we turn to the theory of convex calibrated surrogate losses – which has yielded convex risk minimization algorithms for several other discrete prediction problems in recent years (Bartlett et al., 2006; Zhang, 2004b; Tewari & Bartlett, 2007; Steinwart, 2007; Duchi et al., 2010; Gao & Zhou, 2013; Ramaswamy et al., 2014; 2015) – to design principled surrogate risk minimization algorithms for the multi-label  $F_\beta$ -measure. In particular, for an MLC problem with  $s$  tags, the total number of possible labelings of an instance is  $2^s$  (each tag can be active or inactive). Viewing the  $F_\beta$ -measure as (one minus) a  $2^s \times 2^s$  loss matrix, we show that this matrix has rank at most  $s^2 + 1$ , and apply the results of Ramaswamy et al. (2014) to design an output coding scheme that reduces the  $F_\beta$  learning problem to a set of  $s^2 + 1$  binary class probability estimation (CPE) problems. By using a suitable binary surrogate risk minimization algorithm (such as binary logistic regression) for these binary

---

<sup>1</sup>Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA <sup>2</sup>Department of Computer Science and Engineering, Indian Institute of Technology Madras, Chennai, India. Correspondence to: Shivani Agarwal <ashivani@seas.upenn.edu>.

problems, we effectively construct a  $(s^2 + 1)$ -dimensional convex calibrated surrogate loss for the  $F_\beta$ -measure. We also give a quantitative regret transfer bound for the constructed surrogate, which allows us to transfer any regret guarantees for the binary subproblems to guarantees on  $F_\beta$ -regret for the overall MLC problem. In particular, this means that using a consistent learner for the binary problems yields a consistent learner for the MLC problem (whose  $F_\beta$ -regret goes to zero as the training sample size increases).

Our algorithm is related to the plug-in algorithm of Dembczynski et al. (2013), which also estimates  $s^2 + 1$  statistics of the underlying distribution. Dembczynski et al. (2013) estimate these statistics by reducing the  $F_\beta$  maximization problem to  $s$  multiclass CPE problems, each with at most  $s + 1$  classes (plus one binary CPE problem); we do so by reducing the problem to  $s^2 + 1$  binary CPE problems. As we show, both algorithms effectively estimate the same  $s^2 + 1$  statistics, and indeed, both perform similarly in experiments. Interestingly, the algorithm of Dembczynski et al. (2013), while motivated primarily by the plug-in approach, can also be viewed as minimizing a certain convex calibrated surrogate loss (different from ours); conversely, our algorithm, while motivated primarily by the convex calibrated surrogates approach, can also be viewed as a plug-in algorithm. Our study brings out interesting connections between the two approaches; in addition, to the best of our knowledge, our analysis is the first to provide a quantitative regret transfer bound for calibrated surrogates for the  $F_\beta$ -measure.

**Organization.** Section 2 discusses related work. Section 3 gives preliminaries and background. Section 4 gives our convex calibrated surrogates for the  $F_\beta$ -measure; Section 5 provides a regret transfer bound for them. Section 6 discusses the relationship with the plug-in algorithm of Dembczynski et al. (2013). Section 7 summarizes our experiments.

## 2. Related Work

There has been much work on multi-label learning, learning with the  $F_\beta$ -measure, and convex calibrated surrogates. Below we briefly discuss work that is most related to our study. For detailed surveys on multi-label learning, we refer the reader to Zhang & Zhou (2014) and Pillai et al. (2017).

**Bayes optimal multi-label classifiers.** In an elegant study, Dembczynski et al. (2011) studied in detail the form of a Bayes optimal multi-label classifier for the  $F_1$ -measure. In particular, they showed that, for an  $s$ -label MLC problem, given a certain set of  $s^2 + 1$  statistics of the true conditional label distribution (distribution over  $2^s$  labelings), one can compute a Bayes optimal classifier for the  $F_1$ -measure in  $O(s^3)$  time. Their result extends to general  $F_\beta$ -measures. Bayes optimal classifiers have also been studied for other MLC performance measures, such as Hamming loss and subset 0-1 loss (Dembczynski et al., 2010).

**Consistent algorithms for multi-label learning.** Dembczynski et al. (2013) extended and operationalized the results of Dembczynski et al. (2011) by providing a consistent plug-in MLC algorithm for the  $F_\beta$ -measure. Specifically, they showed that the  $s^2 + 1$  statistics of the conditional label distribution needed to compute a Bayes optimal classifier can be estimated via  $s$  multiclass CPE problems, each with at most  $s + 1$  classes, plus one binary CPE problem; the statistics estimated by solving these CPE problems can then be plugged into the  $O(s^3)$ -time procedure of Dembczynski et al. (2011) to produce a consistent plug-in algorithm termed the *exact F-measure plug-in* (EFP) algorithm. Consistent learning algorithms have also been studied for other multi-label performance measures (Gao & Zhou, 2013; Koyejo et al., 2015).<sup>1</sup> The simple approach of learning an independent binary classifier for each of the  $s$  labels, known as *binary relevance* (BR), is known to yield a consistent algorithm for the Hamming loss; it also yields a consistent algorithm for the  $F_\beta$ -measure under the assumption of conditionally independent labels, but can be arbitrarily bad otherwise (Dembczynski et al., 2011).

**Large-margin algorithms for multi-label learning.** Several studies have considered large-margin algorithms for multi-label learning with the  $F_\beta$ -measure. These include the *reverse multi-label* (RML) and *sub-modular multi-label* (SML) algorithms of Petterson & Caetano (2010; 2011), which make use of the StructSVM framework (Tsochantiridis et al., 2005), and more recently, the *label-wise and instance-wise margin optimization* (LIMO) algorithm due to Wu & Zhou (2017), which aims to simultaneously optimize several different multi-label performance measures. The RML and SML algorithms were proven to be inconsistent for the  $F_\beta$ -measure and shown to be outperformed by the EFP algorithm by Dembczynski et al. (2013). We include a comparison with LIMO in our experiments.

**Multivariate  $F_\beta$ -measure for binary classification.** The  $F_\beta$ -measure is also used as a multivariate performance measure in binary classification tasks with significant class imbalance. This use of the  $F_\beta$ -measure is related to, but distinct from, the use of the  $F_\beta$ -measure in MLC problems. Several approaches have been proposed that aim to optimize the multivariate  $F_\beta$ -measure in binary classification (Joachims, 2005; Ye et al., 2012; Parambath et al., 2014).

<sup>1</sup>Note that while the study of Koyejo et al. (2015) also includes the  $F_\beta$ -measure (among other performance measures), their study is in the context of what has been referred to as the ‘expected utility maximization’ (EUM) framework; in contrast, our study is in the context of what has been referred to as the ‘decision-theoretic analysis’ (DTA) framework. Their results are generally incomparable to ours. (In particular, under the EUM framework, Koyejo et al. (2015) showed that a thresholding approach leads to Bayes optimal performance; on the contrary, under the DTA framework, it was shown by Dembczynski et al. (2011) that a thresholding approach cannot be optimal for general distributions.)

**Convex calibrated surrogates.** Convex surrogate losses are frequently used in machine learning to design computationally efficient learning algorithms. The notion of calibrated surrogate losses, which ensures that minimizing the surrogate loss can (in the limit of sufficient data) recover a Bayes optimal model for the target discrete loss, was initially studied in the context of binary classification (Bartlett et al., 2006; Zhang, 2004a) and multiclass 0-1 classification (Zhang, 2004b; Tewari & Bartlett, 2007). In recent years, calibrated surrogates have been designed for several more complex learning problems, including general multiclass problems and certain types of subset ranking and multi-label problems (Steinwart, 2007; Duchi et al., 2010; Gao & Zhou, 2013; Ramaswamy et al., 2013; 2014; 2015). In our work, we will make use of a result of Ramaswamy et al. (2014), who designed convex calibrated surrogates based on output coding for multiclass problems with low-rank loss matrices.

### 3. Preliminaries and Background

#### 3.1. Problem Setup

**Multi-label classification (MLC).** In an MLC problem, there is an instance space  $\mathcal{X}$ , and a set of  $s$  labels or ‘tags’  $\mathcal{L} = [s] := \{1, \dots, s\}$  that can be associated with each instance in  $\mathcal{X}$ . For example, in image tagging,  $\mathcal{X}$  is the set of possible images, and  $\mathcal{L}$  is a set of  $s$  pre-defined tags (such as `sky`, `cloud`, `water` etc) that can be associated with each image. The learner is given a training sample  $S = \{(x_1, \mathbf{y}_1), \dots, (x_m, \mathbf{y}_m)\} \in (\mathcal{X} \times \{0, 1\}^s)^m$ , where the labeling  $\mathbf{y}_i \in \{0, 1\}^s$  indicates which of the  $s$  tags are active in instance  $x_i$  (specifically,  $y_{ij} = 1$  denotes that tag  $j$  is active in instance  $x_i$ , and  $y_{ij} = 0$  denotes it is inactive). The goal is to learn from these examples a multi-label classifier  $\mathbf{h} : \mathcal{X} \rightarrow \{0, 1\}^s$  which, given a new instance  $x \in \mathcal{X}$ , predicts which tags are active or inactive via  $\mathbf{h}(x) \in \{0, 1\}^s$ .

**$F_\beta$ -measure.** For any  $\beta > 0$ , the  $F_\beta$ -measure evaluates the quality of an MLC prediction as follows. Given a true labeling  $\mathbf{y} \in \{0, 1\}^s$  and a predicted labeling  $\hat{\mathbf{y}} \in \{0, 1\}^s$ , the recall and precision are given by

$$\text{rec}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\sum_{j=1}^s y_j \hat{y}_j}{\|\hat{\mathbf{y}}\|_1}; \quad \text{prec}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\sum_{j=1}^s y_j \hat{y}_j}{\|\mathbf{y}\|_1}.$$

In words, the recall measures the fraction of active tags that are predicted correctly, and the precision measures the fraction of tags predicted as active that are actually so. The  $F_\beta$ -measure balances these two quantities by taking their (weighted) harmonic mean:

$$\begin{aligned} F_\beta(\mathbf{y}, \hat{\mathbf{y}}) &= \left( \left( \frac{\beta^2}{1+\beta^2} \right) \frac{1}{\text{rec}(\mathbf{y}, \hat{\mathbf{y}})} + \left( \frac{1}{1+\beta^2} \right) \frac{1}{\text{prec}(\mathbf{y}, \hat{\mathbf{y}})} \right)^{-1} \\ &= \frac{(1+\beta^2) \sum_{j=1}^s y_j \hat{y}_j}{\beta^2 \|\mathbf{y}\|_1 + \|\hat{\mathbf{y}}\|_1}. \end{aligned} \quad (1)$$

Clearly,  $0 \leq F_\beta(\mathbf{y}, \hat{\mathbf{y}}) \leq 1$ . Higher values of the  $F_\beta$ -

measure correspond to better quality predictions. We will take  $\frac{0}{0} = 1$ , so that when  $\mathbf{y} = \hat{\mathbf{y}} = \mathbf{0}$ , we have  $F_\beta(\mathbf{0}, \mathbf{0}) = 1$ . The most commonly used instantiation is the  $F_1$ -measure, which weighs recall and precision equally; other commonly used variants include the  $F_2$ -measure, which weighs recall more heavily than precision, and the  $F_{0.5}$ -measure, which weighs precision more heavily than recall.

**Learning goal.** Assuming that training examples are drawn IID from some underlying probability distribution  $D$  on  $\mathcal{X} \times \{0, 1\}^s$ , it is natural then to measure the quality of a multi-label classifier  $\mathbf{h} : \mathcal{X} \rightarrow \{0, 1\}^s$  by its  $F_\beta$ -generalization accuracy:<sup>2</sup>

$$\text{acc}_D^{F_\beta}[\mathbf{h}] = \mathbf{E}_{(x, \mathbf{y}) \sim D} [F_\beta(\mathbf{y}, \mathbf{h}(x))].$$

The Bayes  $F_\beta$ -accuracy is then the highest possible value of the  $F_\beta$ -generalization accuracy for  $D$ :

$$\text{acc}_D^{F_\beta, *} = \sup_{\mathbf{h} : \mathcal{X} \rightarrow \{0, 1\}^s} \text{acc}_D^{F_\beta}[\mathbf{h}].$$

The  $F_\beta$ -regret of a multi-label classifier  $\mathbf{h}$  is then the difference between the Bayes  $F_\beta$ -accuracy and the  $F_\beta$ -accuracy of  $\mathbf{h}$ :

$$\text{regret}_D^{F_\beta}[\mathbf{h}] = \text{acc}_D^{F_\beta, *} - \text{acc}_D^{F_\beta}[\mathbf{h}].$$

Our goal will be to design *consistent* algorithms for the  $F_\beta$ -measure, i.e. algorithms whose  $F_\beta$ -regret converges (in probability) to zero as the number of training examples increases. In particular, since we cannot maximize the (discrete)  $F_\beta$ -measure directly, we would like to design consistent algorithms that maximize a concave surrogate performance measure – or equivalently, minimize a convex surrogate loss – instead. For this, we will turn to the theory of convex calibrated surrogates.

#### 3.2. Convex Calibrated Surrogates for Multiclass Problems

Here we review the theory of convex calibrated surrogates for multiclass classification problems, and in particular, the result of Ramaswamy et al. (2014) for low-rank multiclass loss matrices that we will use in our work. We will apply the theory to the multi-label  $F_\beta$ -measure in Section 4.

**Multiclass classification.** Consider a standard multiclass (not multi-label) learning problem with instance space  $\mathcal{X}$  and label space  $\mathcal{Y} = [n]$  (i.e.,  $n$  classes). Let  $\mathbf{L} \in \mathbb{R}_+^{n \times n}$  be a loss matrix whose  $(y, \hat{y})$ -th entry  $\ell_{y, \hat{y}} = \ell(y, \hat{y})$  (for each  $y, \hat{y} \in [n]$ ) specifies the loss incurred on predicting  $\hat{y}$  when the true label is  $y$  (the 0-1 loss  $\mathbf{L}^{0-1}$  is a special case with  $\ell_{y, \hat{y}}^{0-1} = \mathbf{1}(\hat{y} \neq y)$ ). Then, given a training sample  $S = ((x_1, y_1), \dots, (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m$  with examples drawn IID from some underlying probability distribution  $D$  on  $\mathcal{X} \times \mathcal{Y}$ , the performance of a classifier  $h : \mathcal{X} \rightarrow \mathcal{Y}$  is measured by its  $\mathbf{L}$ -generalization error  $\text{er}_D^{\mathbf{L}}[h] = \mathbf{E}_{(x, y) \sim D} [\ell_{y, h(x)}]$ ,

<sup>2</sup>Note that our focus is on *instance-averaged*  $F_\beta$  performance (Zhang & Zhou, 2014).

or its  $\mathbf{L}$ -regret  $\text{regret}_D^{\mathbf{L}}[h] = \text{er}_D^{\mathbf{L}}[h] - \text{er}_D^{\mathbf{L},*}$ , where  $\text{er}_D^{\mathbf{L},*} = \inf_{h: \mathcal{X} \rightarrow \mathcal{Y}} \text{er}_D^{\mathbf{L}}[h]$  is the Bayes  $\mathbf{L}$ -error for  $D$ . A learning algorithm that maps training samples  $S$  to classifiers  $h_S$  is said to be (universally)  $\mathbf{L}$ -consistent if for all  $D$  and for  $S \sim D^m$ ,  $\text{regret}_D^{\mathbf{L}}[h_S] \xrightarrow{P} 0$  as  $m \rightarrow \infty$ .

**Surrogate risk minimization and calibrated surrogates.** Since minimizing the discrete loss  $\mathbf{L}$  directly is computationally hard, a common algorithmic framework is to minimize a surrogate loss  $\psi: [n] \times \mathbb{R}^d \rightarrow \mathbb{R}_+$  for some suitable  $d \in \mathbb{Z}_+$ . In particular, given a multiclass training sample  $S$  as above, one learns a  $d$ -dimensional ‘scoring’ function  $\mathbf{f}_S: \mathcal{X} \rightarrow \mathbb{R}^d$  by solving

$$\min_{\mathbf{f}} \sum_{i=1}^m \psi(y_i, \mathbf{f}(x_i))$$

over a suitably rich class of functions  $\mathbf{f}: \mathcal{X} \rightarrow \mathbb{R}^d$ ; and then returns  $h_S = \text{decode} \circ \mathbf{f}_S$  for some suitable mapping  $\text{decode}: \mathbb{R}^d \rightarrow [n]$ . In practice, the surrogate  $\psi$  is often chosen to be convex in its second argument to enable efficient minimization. It is known that if the minimization is performed over a universal function class (with suitable regularization), then the resulting algorithm is universally  $\psi$ -consistent, i.e. that the  $\psi$ -regret converges to zero:  $\text{regret}_D^{\psi}[\mathbf{f}_S] = \text{er}_D^{\psi}[\mathbf{f}_S] - \text{er}_D^{\psi,*} \xrightarrow{P} 0$  as  $m \rightarrow \infty$  (where  $\text{er}_D^{\psi}[\mathbf{f}] = \mathbb{E}_{(x,y) \sim D}[\psi(y, \mathbf{f}(x))]$  is the  $\psi$ -generalization error of  $\mathbf{f}$  and  $\text{er}_D^{\psi,*} = \inf_{\mathbf{f}: \mathcal{X} \rightarrow \mathbb{R}^d} \text{er}_D^{\psi}[\mathbf{f}]$  is the Bayes  $\psi$ -error). The surrogate  $\psi$ , together with the mapping  $\text{decode}$ , is said to be  $\mathbf{L}$ -calibrated if this also implies  $\mathbf{L}$ -consistency, i.e. if

$$\text{regret}_D^{\psi}[\mathbf{f}_S] \xrightarrow{P} 0 \implies \text{regret}_D^{\mathbf{L}}[\text{decode} \circ \mathbf{f}_S] \xrightarrow{P} 0.$$

Thus, given a target loss  $\mathbf{L}$ , the task of designing an  $\mathbf{L}$ -consistent algorithm reduces to designing a convex  $\mathbf{L}$ -calibrated surrogate-mapping pair  $(\psi, \text{decode})$ ; the resulting surrogate risk minimization algorithm (implemented in a universal function class with suitable regularization) is then universally  $\mathbf{L}$ -consistent.

**Result of Ramaswamy et al. (2014) for low-rank loss matrices.** The result of Ramaswamy et al. (2014) effectively decomposes multiclass problems into a set of binary CPE problems; to describe the result, we will need the following definition for *binary* losses:

**Definition 1** (Strictly proper composite binary losses (Reid & Williamson, 2010)). A binary loss  $\phi: \{\pm 1\} \times \mathbb{R} \rightarrow \mathbb{R}_+$  is strictly proper composite with underlying (invertible) link function  $\gamma: [0, 1] \rightarrow \mathbb{R}$  if for all  $q \in [0, 1]$  and  $u \neq \gamma(q) \in \mathbb{R}$ :

$$\mathbb{E}_{y \sim \text{Bin}^{\pm 1}(q)} [\phi(y, u) - \phi(y, \gamma(q))] > 0,$$

where  $y \sim \text{Bin}^{\pm 1}(q)$  denotes a  $\{\pm 1\}$ -valued random variable that takes value  $+1$  with probability  $q$  and value  $-1$  with probability  $1 - q$ .

Intuitively, minimizing a strictly proper composite binary loss allows one to recover accurate class probability estimates for binary CPE problems: the learned real-valued score is simply inverted via  $\gamma^{-1}$  (Reid & Williamson, 2010).

We can now state the result of Ramaswamy et al. (2014), which for multiclass loss matrices  $\mathbf{L}$  of rank  $r$ , gives a family of  $r$ -dimensional convex  $\mathbf{L}$ -calibrated surrogates defined in terms of strictly proper composite binary losses as follows (result specialized here to the case of square loss matrices, and stated with a small change in normalization):

**Theorem 2** (Ramaswamy et al. (2014)). Let  $\mathbf{L} \in \mathbb{R}_+^{n \times n}$  be a rank- $r$  multiclass loss matrix, with  $\ell_{y,\hat{y}} = \mathbf{a}_y^\top \mathbf{b}_{\hat{y}}$  for some  $\mathbf{a}_1, \dots, \mathbf{a}_n, \mathbf{b}_1, \dots, \mathbf{b}_n \in \mathbb{R}^r$ . Let  $\phi: \{\pm 1\} \times \mathbb{R} \rightarrow \mathbb{R}_+$  be any strictly proper composite binary loss, with underlying link function  $\gamma: [0, 1] \rightarrow \mathbb{R}$ . Define a multiclass surrogate  $\psi: [n] \times \mathbb{R}^r \rightarrow \mathbb{R}_+$  and mapping  $\text{decode}: \mathbb{R}^r \rightarrow [n]$  as follows:

$$\begin{aligned} \psi(y, \mathbf{u}) &= \sum_{j=1}^r \left( \tilde{a}_{yj} \phi(+1, u_j) + (1 - \tilde{a}_{yj}) \phi(-1, u_j) \right) \\ \text{decode}(\mathbf{u}) &\in \underset{\hat{y} \in [n]}{\text{argmin}} \sum_{j=1}^r \tilde{b}_{\hat{y}j} \gamma^{-1}(u_j) + c_{\hat{y}}, \end{aligned}$$

where

$$\begin{aligned} \tilde{a}_{yj} &= \frac{a_{yj} - a_{\min}}{a_{\max} - a_{\min}} \quad (\in [0, 1]) \\ \tilde{b}_{\hat{y}j} &= (a_{\max} - a_{\min}) \cdot b_{\hat{y}j} \\ c_{\hat{y}} &= a_{\min} \sum_{j=1}^r b_{\hat{y}j} \\ a_{\min} &= \min_{y,j} a_{yj} \\ a_{\max} &= \max_{y,j} a_{yj}. \end{aligned}$$

Then  $(\psi, \text{decode})$  is  $\mathbf{L}$ -calibrated.

The above result effectively decomposes the multiclass problem into  $r$  binary CPE problems, where the labels for these CPE problems can themselves be given as probabilities in  $[0, 1]$  rather than binary values (see Ramaswamy et al. (2014) for details). For our purposes, we will use the standard binary logistic loss for the binary CPE problems, which is known to be strictly proper composite (see Section 4 below for more details).

## 4. Convex Calibrated Surrogates for $F_\beta$

In order to construct convex calibrated surrogates – and corresponding surrogate risk minimization algorithms – for the multi-label  $F_\beta$ -measure, we will start by viewing the multi-label learning problem as a giant multiclass classification problem with  $n = 2^s$  classes (this is only for the purpose of analysis and derivation of the surrogates; as we will see, the actual algorithms we will obtain will require learning only  $O(s^2)$  real-valued score functions). To this end, let us define the  $F_\beta$ -loss matrix  $\mathbf{L}^{F_\beta} \in \mathbb{R}_+^{\{0,1\}^s \times \{0,1\}^s}$  as follows:

$$\ell_{\mathbf{y}, \hat{\mathbf{y}}}^{F_\beta} = 1 - F_\beta(\mathbf{y}, \hat{\mathbf{y}}).$$

**$\mathbf{L}^{F_\beta}$  has low rank.** We show here that (a slightly shifted version of) the above loss matrix has rank at most  $s^2 + 1$ .

**Proposition 3.**  $\text{rank}(\mathbf{L}^{F_\beta} - 1) \leq s^2 + 1$ .



*Proof.* We have,

$$\ell_{\mathbf{y}, \hat{\mathbf{y}}}^{F_\beta} - 1 = -F_\beta(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{(1 + \beta^2) \sum_{j=1}^s y_j \hat{y}_j}{\beta^2 \|\mathbf{y}\|_1 + \|\hat{\mathbf{y}}\|_1}.$$

Stratifying over the  $s + 1$  different values of  $\|\mathbf{y}\|_1 \in \{0, 1, \dots, s\}$ , we can write this as

$$\begin{aligned} \ell_{\mathbf{y}, \hat{\mathbf{y}}}^{F_\beta} - 1 &= -\mathbf{1}(\|\mathbf{y}\|_1 = 0) \cdot \mathbf{1}(\|\hat{\mathbf{y}}\|_1 = 0) \\ &\quad - \sum_{k=1}^s \mathbf{1}(\|\mathbf{y}\|_1 = k) \cdot \frac{(1 + \beta^2) \sum_{j=1}^s y_j \hat{y}_j}{\beta^2 k + \|\hat{\mathbf{y}}\|_1} \\ &= a_{\mathbf{y}, 0} \cdot b_{\hat{\mathbf{y}}, 0} + \sum_{j=1}^s \sum_{k=1}^s a_{\mathbf{y}, jk} \cdot b_{\hat{\mathbf{y}}, jk}, \end{aligned}$$

where

$$a_{\mathbf{y}, 0} = \mathbf{1}(\|\mathbf{y}\|_1 = 0) \quad (2)$$

$$b_{\hat{\mathbf{y}}, 0} = -\mathbf{1}(\|\hat{\mathbf{y}}\|_1 = 0) \quad (3)$$

$$a_{\mathbf{y}, jk} = \mathbf{1}(\|\mathbf{y}\|_1 = k) \cdot y_j \quad (4)$$

$$b_{\hat{\mathbf{y}}, jk} = -\frac{(1 + \beta^2) \cdot \hat{y}_j}{\beta^2 k + \|\hat{\mathbf{y}}\|_1}. \quad (5)$$

This proves the claim.  $\square$

**$\mathbf{L}^{F_\beta}$ -calibrated surrogates.** Given the above result, we can now apply Theorem 2 to construct a family of  $(s^2 + 1)$ -dimensional convex calibrated surrogate losses for  $\mathbf{L}^{F_\beta}$ .<sup>3</sup> Specifically, starting with any strictly proper composite binary loss  $\phi : \{\pm 1\} \times \mathbb{R} \rightarrow \mathbb{R}_+$  with underlying link function  $\gamma : [0, 1] \rightarrow \mathbb{R}$ , we define a multiclass surrogate  $\psi : \{0, 1\}^s \times \mathbb{R}^{s^2+1} \rightarrow \mathbb{R}_+$  and mapping  $\text{decode} : \mathbb{R}^{s^2+1} \rightarrow \{0, 1\}^s$  as follows (where we denote  $\mathbf{u} = (u_0, (u_{jk})_{j,k=1}^s)^\top \in \mathbb{R}^{s^2+1}$ ):

$$\begin{aligned} \psi(\mathbf{y}, \mathbf{u}) &= a_{\mathbf{y}, 0} \cdot \phi(+1, u_0) + (1 - a_{\mathbf{y}, 0}) \cdot \phi(-1, u_0) \\ &\quad + \sum_{j=1}^s \sum_{k=1}^s a_{\mathbf{y}, jk} \cdot \phi(+1, u_{jk}) + (1 - a_{\mathbf{y}, jk}) \cdot \phi(-1, u_{jk}) \end{aligned} \quad (6)$$

$\text{decode}(\mathbf{u})$

$$\in \underset{\hat{\mathbf{y}} \in \{0, 1\}^s}{\text{argmin}} \quad b_{\hat{\mathbf{y}}, 0} \cdot \gamma^{-1}(u_0) + \sum_{j=1}^s \sum_{k=1}^s b_{\hat{\mathbf{y}}, jk} \cdot \gamma^{-1}(u_{jk}), \quad (7)$$

where  $a_{\mathbf{y}, 0}, a_{\mathbf{y}, jk}, b_{\hat{\mathbf{y}}, 0}, b_{\hat{\mathbf{y}}, jk}$  are as defined in Eqs. (2-5). Then, by Theorem 2 and the proof of Proposition 3, it follows that  $(\psi, \text{decode})$  is  $\mathbf{L}^{F_\beta}$ -calibrated.<sup>4</sup> Therefore, the resulting  $(\psi, \text{decode})$ -based surrogate risk minimization algorithm, when implemented in a universal function class (with

<sup>3</sup>Note that minimizing the  $\mathbf{L}^{F_\beta}$ -generalization error is equivalent to minimizing the  $(\mathbf{L}^{F_\beta} - 1)$ -generalization error, and therefore a calibrated surrogate for  $\mathbf{L}^{F_\beta} - 1$  is also calibrated for  $\mathbf{L}^{F_\beta}$ .

<sup>4</sup>Note that when applying Theorem 2 here, we have  $a_{\min} = 0$  and  $a_{\max} = 1$ , and therefore  $\tilde{\mathbf{a}}_{\mathbf{y}} = \mathbf{a}_{\mathbf{y}}$ ,  $\tilde{\mathbf{b}}_{\hat{\mathbf{y}}} = \mathbf{b}_{\hat{\mathbf{y}}}$ , and  $c_{\hat{\mathbf{y}}} = 0$ .

**Algorithm 1** Surrogate risk minimization algorithm for multi-label  $F_\beta$ -measure

- 1: **Input:** Training sample  $S = ((x_1, \mathbf{y}_1), \dots, (x_m, \mathbf{y}_m)) \in (\mathcal{X} \times \{0, 1\}^s)^m$
- 2: **Parameters:** (1) Strictly proper composite binary CPE loss  $\phi : \{\pm 1\} \times \mathbb{R} \rightarrow \mathbb{R}_+$ ; (2) Class  $\mathcal{F}$  of functions  $\mathbf{f} : \mathcal{X} \rightarrow \mathbb{R}^{s^2+1}$
- 3: Find  $\mathbf{f}_S \in \underset{\mathbf{f} \in \mathcal{F}}{\text{argmin}} \sum_{i=1}^m \psi(\mathbf{y}_i, \mathbf{f}(x_i))$ , where  $\psi$  is as defined in Eq. (6)
- 4: **Output:** Multi-label classifier  $\mathbf{h}_S = \text{decode} \circ \mathbf{f}_S$ , where  $\text{decode}$  is as defined in Eq. (7) (see Appendix for efficient implementation of  $\text{decode}$ )

suitable regularization), is consistent for the  $F_\beta$ -measure. The algorithm is summarized in Algorithm 2. Note that since  $a_{\mathbf{y}, 0}, a_{\mathbf{y}, jk} \in \{0, 1\}$ , in this case minimizing the surrogate risk above amounts to solving  $s^2 + 1$  binary CPE problems with standard binary (non-probabilistic) labels.

**Choice of strictly proper composite binary loss  $\phi$ .** As a specific instantiation, in our experiments, we will make use of the binary logistic loss  $\phi_{\log} : \{\pm 1\} \times \mathbb{R} \rightarrow \mathbb{R}_+$  given by

$$\phi_{\log}(y, u) = \ln(1 + e^{-yu}) \quad (8)$$

as the binary loss above; this is known to be strictly proper composite (Reid & Williamson, 2010), with underlying logit link function  $\gamma_{\log} : [0, 1] \rightarrow \mathbb{R}$  given by

$$\gamma_{\log}(p) = \ln\left(\frac{p}{1-p}\right). \quad (9)$$

**Implementation of ‘decode’ mapping.** The mapping  $\text{decode} : \mathbb{R}^{s^2+1} \rightarrow \{0, 1\}^s$  above can be implemented in  $O(s^3)$  time using a procedure due to Dembczynski et al. (2011); details are provided in the Appendix for completeness. In particular, Dembczynski et al. (2011) show that if one knows the true conditional MLC distribution  $p(\mathbf{y}|x)$ , then one can use  $s^2 + 1$  statistics of this distribution to construct a Bayes optimal classifier for the  $F_\beta$ -measure; they then provide a procedure to perform this computation in  $O(s^3)$  time. As we discuss in greater detail in Section 6, our surrogate loss  $\psi$  can be viewed as computing estimates of the same  $s^2 + 1$  statistics from the training sample  $S$ , and therefore our algorithm, which applies the ‘decoding’ procedure of Dembczynski et al. (2011) to these estimated quantities, can be viewed as effectively learning a form of ‘plug-in’ multi-label classifier for the  $F_\beta$ -measure.

## 5. Regret Transfer Bound

Above, we constructed a family of  $\mathbf{L}^{F_\beta}$ -calibrated surrogate-mapping pairs  $(\psi, \text{decode})$  (Eqs. (6-7)), yielding a family of surrogate risk minimization algorithms for the  $F_\beta$ -measure (Algorithm 2). We now give a quantitative regret

transfer bound showing that any guarantees on the surrogate  $\psi$ -regret also translate to guarantees on the target  $F_\beta$ -regret. Specifically, the surrogate loss  $\psi$  was defined in terms of a constituent strictly proper composite binary loss  $\phi : \{\pm 1\} \times \mathbb{R} \rightarrow \mathbb{R}_+$ . We show that if the binary loss  $\phi$  is *strongly proper composite* (a relatively mild condition satisfied by several common strictly proper composite binary losses, including the logistic loss), then for all models  $\mathbf{f} : \mathcal{X} \rightarrow \mathbb{R}^{s^2+1}$ , we can upper bound  $\text{regret}_D^{F_\beta}[\text{decode} \circ \mathbf{f}]$ , the target  $F_\beta$ -regret of the multi-label classifier given by  $\mathbf{h}(x) = \text{decode}(\mathbf{f}(x))$ , in terms of  $\text{regret}_D^\psi[\mathbf{f}]$ , the surrogate regret of  $\mathbf{f}$ . In order to prove the regret transfer bound, we will need the following definition:

**Definition 4** (Strongly proper composite binary losses (Agarwal, 2014)). *Let  $\lambda > 0$ . A binary loss  $\phi : \{\pm 1\} \times \mathbb{R} \rightarrow \mathbb{R}_+$  is said to be  $\lambda$ -strongly proper composite with underlying (invertible) link function  $\gamma : [0, 1] \rightarrow \mathbb{R}$  if for all  $q \in [0, 1]$ ,  $u \in \mathbb{R}$ :*

$$\mathbf{E}_{y \sim \text{Bin}^{\pm 1}(q)} [\phi(y, u) - \phi(y, \gamma(q))] \geq \frac{\lambda}{2} (\gamma^{-1}(u) - q)^2.$$

We note that the logistic loss (Eq. (8)) is known to be 4-strongly proper composite with underlying link given by the logit link (Eq. (9)) (Agarwal, 2014).

**Additional notation.** To prove our regret transfer bound, we will also need some additional notation. In particular, for each  $\mathbf{y}, \hat{\mathbf{y}} \in \{0, 1\}^s$ , we will define the vectors

$$\mathbf{a}_{\mathbf{y}} = \begin{pmatrix} a_{\mathbf{y},0} \\ a_{\mathbf{y},11} \\ \vdots \\ a_{\mathbf{y},ss} \end{pmatrix} \in \{0, 1\}^{s^2+1} \quad (10)$$

$$\mathbf{b}_{\hat{\mathbf{y}}} = \begin{pmatrix} b_{\hat{\mathbf{y}},0} \\ b_{\hat{\mathbf{y}},11} \\ \vdots \\ b_{\hat{\mathbf{y}},ss} \end{pmatrix} \in \mathbb{R}^{s^2+1}, \quad (11)$$

where  $a_{\mathbf{y},0}, a_{\mathbf{y},jk}, b_{\hat{\mathbf{y}},0}, b_{\hat{\mathbf{y}},jk}$  are as defined in Eqs. (2-5). Moreover, for each  $x \in \mathcal{X}$ , we will define

$$\mathbf{q}(x) = \mathbf{E}_{\mathbf{y}|x}[\mathbf{a}_{\mathbf{y}}] = \sum_{\mathbf{y} \in \{0,1\}^s} p(\mathbf{y}|x) \cdot \mathbf{a}_{\mathbf{y}} \in [0, 1]^{s^2+1}. \quad (12)$$

Intuitively, the elements  $q_0(x), (q_{jk}(x))_{j,k=1}^s$  of  $\mathbf{q}(x)$  are the ‘class probability functions’ corresponding to the  $s^2 + 1$  binary CPE problems effectively created by the surrogate loss  $\psi$  defined in Eq. (6). The function  $\mathbf{f}_S : \mathcal{X} \rightarrow \mathbb{R}^{s^2+1}$  learned by minimizing  $\psi$  will be such that  $\gamma^{-1}(\mathbf{f}_S(x))$  will serve as an estimate of  $\mathbf{q}(x)$ .

**Regret transfer bound.** We are now ready to state and prove the following regret transfer bound for the family of surrogate losses defined in the previous section:

**Theorem 5.** *Let  $\phi : \{\pm 1\} \times \mathbb{R} \rightarrow \mathbb{R}_+$  be a  $\lambda$ -strongly proper composite binary loss with underlying link function  $\gamma : [0, 1] \rightarrow \mathbb{R}$ . Let  $(\psi, \text{decode})$  be defined as in Eqs. (6-7). Then for all probability distributions  $D$  on  $\mathcal{X} \times \{0, 1\}^s$  and all  $\mathbf{f} : \mathcal{X} \rightarrow \mathbb{R}^{s^2+1}$ , we have*

$$\text{regret}_D^{F_\beta}[\text{decode} \circ \mathbf{f}] \leq \frac{1 + \beta^2}{\beta} \sqrt{\frac{2(\ln s + 1)}{\lambda}} \cdot \text{regret}_D^\psi[\mathbf{f}].$$

*Proof.* We have,

$$\begin{aligned} & \text{regret}_D^{F_\beta}[\text{decode} \circ \mathbf{f}] \\ &= \mathbf{E}_x \left[ \sum_{\mathbf{y}} p(\mathbf{y}|x) \cdot \left( \ell_{\mathbf{y}, \text{decode}(\mathbf{f}(x))}^{F_\beta} - \min_{\hat{\mathbf{y}}} \ell_{\mathbf{y}, \hat{\mathbf{y}}}^{F_\beta} \right) \right] \\ &= \mathbf{E}_x \left[ \sum_{\mathbf{y}} p(\mathbf{y}|x) \cdot \left( \mathbf{a}_{\mathbf{y}}^\top \mathbf{b}_{\text{decode}(\mathbf{f}(x))} - \min_{\hat{\mathbf{y}}} \mathbf{a}_{\mathbf{y}}^\top \mathbf{b}_{\hat{\mathbf{y}}} \right) \right] \\ &= \mathbf{E}_x \left[ \mathbf{q}(x)^\top \mathbf{b}_{\text{decode}(\mathbf{f}(x))} - \min_{\hat{\mathbf{y}}} \mathbf{q}(x)^\top \mathbf{b}_{\hat{\mathbf{y}}} \right] \\ &= \mathbf{E}_x \left[ \max_{\hat{\mathbf{y}}} \mathbf{q}(x)^\top \left( \mathbf{b}_{\text{decode}(\mathbf{f}(x))} - \mathbf{b}_{\hat{\mathbf{y}}} \right) \right] \\ &\leq \mathbf{E}_x \left[ \max_{\hat{\mathbf{y}}} \left( \mathbf{q}(x) - \gamma^{-1}(\mathbf{f}(x)) \right)^\top \left( \mathbf{b}_{\text{decode}(\mathbf{f}(x))} - \mathbf{b}_{\hat{\mathbf{y}}} \right) \right] \\ &\quad \text{(since by the definition of decode,} \\ &\quad \quad -\gamma^{-1}(\mathbf{f}(x))^\top (\mathbf{b}_{\text{decode}(\mathbf{f}(x))} - \mathbf{b}_{\hat{\mathbf{y}}}) \geq 0 \ \forall \hat{\mathbf{y}}) \\ &\leq \mathbf{E}_x \left[ \left\| \mathbf{q}(x) - \gamma^{-1}(\mathbf{f}(x)) \right\|_2 \cdot \max_{\hat{\mathbf{y}}} \left\| \mathbf{b}_{\text{decode}(\mathbf{f}(x))} - \mathbf{b}_{\hat{\mathbf{y}}} \right\|_2 \right] \\ &\quad \text{(by the Cauchy-Schwarz inequality)} \\ &\leq 2 \max_{\hat{\mathbf{y}}} \left\| \mathbf{b}_{\hat{\mathbf{y}}} \right\|_2 \cdot \mathbf{E}_x \left[ \left\| \mathbf{q}(x) - \gamma^{-1}(\mathbf{f}(x)) \right\|_2 \right]. \quad (13) \end{aligned}$$

Now, since  $\phi$  is  $\lambda$ -strongly proper composite with link function  $\gamma$ , we have

$$\begin{aligned} & \mathbf{E}_x \left[ \left\| \mathbf{q}(x) - \gamma^{-1}(\mathbf{f}(x)) \right\|_2^2 \right] \\ &= \mathbf{E}_x \left[ \left( q_0(x) - \gamma^{-1}(f_0(x)) \right)^2 + \sum_{j=1}^s \sum_{k=1}^s \left( q_{jk}(x) - \gamma^{-1}(f_{jk}(x)) \right)^2 \right] \\ &\leq \frac{2}{\lambda} \mathbf{E}_x \left[ \mathbf{E}_{y \sim \text{Bin}^{\pm 1}(q_0(x))} [\phi(y, f_0(x)) - \phi(y, \gamma(q_0(x)))] + \sum_{j=1}^s \sum_{k=1}^s \mathbf{E}_{y \sim \text{Bin}^{\pm 1}(q_{jk}(x))} [\phi(y, f_{jk}(x)) - \phi(y, \gamma(q_{jk}(x)))] \right] \\ &\quad \text{(by } \lambda\text{-strong proper compositeness of } \phi) \\ &= \frac{2}{\lambda} \mathbf{E}_x \left[ \mathbf{E}_{\mathbf{y}|x} [\psi(\mathbf{y}, \mathbf{f}(x)) - \inf_{\mathbf{u} \in \mathbb{R}^{s^2+1}} \psi(\mathbf{y}, \mathbf{u})] \right] \\ &= \frac{2}{\lambda} \text{regret}_D^\psi[\mathbf{f}]. \quad (14) \end{aligned}$$

Moreover, we have

$$\|\mathbf{b}_0\| = 1,$$

and for  $\hat{\mathbf{y}} \neq \mathbf{0}$ , we have

$$\|\mathbf{b}_{\hat{\mathbf{y}}}\|_2^2 = (1 + \beta^2)^2 \sum_{k=1}^s g_k(\|\hat{\mathbf{y}}\|_1),$$

where

$$g_k(t) = \frac{t}{(\beta^2 k + t)^2}.$$

It can be verified that  $g_k(t)$  is maximized at  $t^* = \beta^2 k$ , yielding for each  $\hat{\mathbf{y}} \neq \mathbf{0}$ ,

$$\begin{aligned} \|\mathbf{b}_{\hat{\mathbf{y}}}\|_2^2 &\leq (1 + \beta^2)^2 \sum_{k=1}^s g_k(\beta^2 k) \\ &= (1 + \beta^2)^2 \sum_{k=1}^s \frac{1}{4\beta^2 k} \\ &\leq \frac{(1 + \beta^2)^2}{4\beta^2} (\ln s + 1) \\ &\quad (\text{since } \sum_{k=1}^s \frac{1}{k} \leq \ln s + 1). \end{aligned}$$

This gives

$$\max_{\hat{\mathbf{y}}} \|\mathbf{b}_{\hat{\mathbf{y}}}\|_2 \leq \frac{(1 + \beta^2)}{2\beta} \sqrt{\ln s + 1}.$$

Combining Eqs. (13-15) and applying Jensen's inequality (to the convex function  $g(z) = z^2$ ) proves the claim.  $\square$

**Remark.** We note that Theorem 5 gives a self-contained proof that the surrogate-mapping pair  $(\psi, \text{decode})$  defined in Eqs. (6-7) is  $\mathbf{L}^{F_\beta}$ -calibrated, since the result implies that for any sequence of models  $\mathbf{f}_S$  learned from training samples  $S \sim D^m$  of increasing size  $m$ ,

$$\text{regret}_D^\psi[\mathbf{f}_S] \xrightarrow{P} 0 \implies \text{regret}_D^{F_\beta}[\text{decode} \circ \mathbf{f}_S] \xrightarrow{P} 0.$$

Nevertheless, since the design of our surrogate-mapping pair  $(\psi, \text{decode})$  was based on the work of Ramaswamy et al. (2014), we chose to present their calibration result (Theorem 2) first. We also note that, while we have stated the above regret transfer bound for the  $F_\beta$ -measure, a similar bound also applies more generally to all multiclass problems with low-rank matrices as considered in Theorem 2, thus yielding a stronger (quantitative) result than Theorem 2 (Ramaswamy, 2015).

## 6. Relationship with Plug-in Algorithm of Dembczynski et al. (2013)

The plug-in algorithm of Dembczynski et al. (2013), termed *exact F-measure plug-in* (EFP), estimates the following statistics of the conditional label distribution  $p(\mathbf{y}|x)$ :

$$\begin{aligned} &\mathbf{P}(\|\mathbf{y}\|_1 = 0 | x) \\ &\mathbf{P}(\|\mathbf{y}\|_1 = k, y_j = 1 | x), \mathbf{P}(y_j = 0 | x), \quad j, k \in [s]. \end{aligned}$$

It formulates estimation of the first statistic above as a binary CPE problem (solved via binary logistic regression), and estimation of the remaining statistics as  $s$  multiclass CPE problems (one for each  $j \in [s]$ ), each with  $s + 1$  classes

(solved via multiclass logistic regression). In practice, since the label vectors  $\mathbf{y}$  are typically sparse (only a small subset of the  $s$  labels are active in any instance), the effective number of classes for each of the  $s$  problems is much smaller than  $s + 1$ , and Dembczynski et al. (2013) exploit this fact by considering the statistics  $\mathbf{P}(\|\mathbf{y}\|_1 = k, y_j = 1 | x)$  only for small  $k$  (based on the maximum number of active labels in the training instances).

As the proof of Theorem 5 makes clear, our algorithm can be viewed as estimating the vector  $\mathbf{q}(x) \in [0, 1]^{s^2+1}$ , with estimation of each component formulated as a binary CPE problem; in particular, having learned a score vector  $\mathbf{f}_S : \mathcal{X} \rightarrow \mathbb{R}^{s^2+1}$ , our algorithm yields  $\gamma^{-1}(\mathbf{f}_S(x)) \in [0, 1]^{s^2+1}$  as an estimate for  $\mathbf{q}(x)$ . A closer look reveals that  $\mathbf{q}(x)$  captures essentially the same  $s^2 + 1$  statistics as above:<sup>5</sup>

$$\begin{aligned} q_0(x) &= \mathbf{E}_{\mathbf{y}|x}[a_{\mathbf{y},0}] = \mathbf{P}(\|\mathbf{y}\|_1 = 0 | x) \\ q_{jk}(x) &= \mathbf{E}_{\mathbf{y}|x}[a_{\mathbf{y},jk}] = \mathbf{P}(\|\mathbf{y}\|_1 = k, y_j = 1 | x), \quad j, k \in [s]. \end{aligned}$$

Thus, both algorithms effectively estimate the same statistics of the conditional label distribution  $p(\mathbf{y}|x)$ ; indeed, these are precisely the statistics needed to compute a Bayes optimal multi-label classifier for the  $F_\beta$ -measure (Dembczynski et al., 2011). In practice, as with the EFP algorithm, our algorithm can also be implemented to estimate  $q_{jk}(x)$  only for small values of  $k$  (i.e. values of  $k$  for which labelings  $\mathbf{y}$  with  $\|\mathbf{y}\|_1 = k$  are actually seen in the training data).

## 7. Experiments

We conducted two sets of experiments to evaluate our algorithm. In the first experiment, we generated synthetic data from a known distribution for which the Bayes optimal  $F_1$ -accuracy could be estimated, and tested the convergence of our algorithm to this optimal  $F_1$  performance. In the second set of experiments, we compared the performance of our algorithm to that of other algorithms on various benchmark data sets. We summarize both sets of experiments below.

### 7.1. Synthetic Data: Convergence to Bayes Optimal $F_1$

In the first experiment, we tested the consistency behavior of our algorithm on a synthetic data set from a known distribution for which the Bayes optimal  $F_1$  performance could be estimated. Specifically, we generated a multi-label data set with instances  $\mathbf{x}$  in  $\mathcal{X} = \mathbb{R}^{100}$  and  $s = 6$  labels/tags (i.e., labelings  $\mathbf{y}$  in  $\{0, 1\}^6$ ), such that the vector  $\mathbf{q}(\mathbf{x}) \in [0, 1]^{37}$  containing the  $s^2 + 1 = 37$  statistics of the conditional label distribution  $p(\mathbf{y}|x)$  needed to compute a Bayes optimal multi-label classifier for  $F_1$  (see Eq. (12)) could be obtained from a linear function of  $\mathbf{x}$ . More precisely, we fixed a matrix  $\mathbf{W} \in [0, 1]^{37 \times 100}$  with entries

<sup>5</sup>Note that for each  $j \in [s]$ , the  $s + 1$  probabilities  $\mathbf{P}(\|\mathbf{y}\|_1 = k, y_j = 1 | x)$  ( $k \in [s]$ ) and  $\mathbf{P}(y_j = 0 | x)$  estimated by the  $j$ -th multiclass problem in EFP add up to 1, so the EFP algorithm effectively estimates a total of  $s^2 + 1$  statistics.

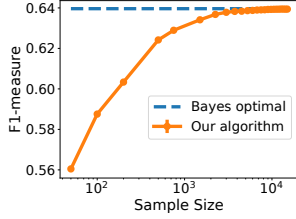


Figure 1. Convergence of our algorithm to Bayes optimal  $F_1$  performance on synthetic multi-label data (see Section 7.1).

drawn uniformly at random from  $[0, 1]$ ; we checked that  $\mathbf{W}$  has full row rank. We also fixed a vector  $\alpha \in [0.1, 1]^{64}$  with entries drawn uniformly from  $[0.1, 1]$ . To generate a data point  $(\mathbf{x}, \mathbf{y})$ , we then did the following: we first sampled  $\mathbf{p} \in \Delta_{64} \equiv \Delta_{\{0,1\}^6}$  from  $\text{Dirichlet}(\alpha)$ . We set  $\mathbf{q} = \mathbf{E}_{\mathbf{y} \sim \mathbf{p}}[\mathbf{a}_{\mathbf{y}}] \in [0, 1]^{37}$ , where  $\mathbf{a}_{\mathbf{y}} \in \{0, 1\}^{37}$  is as defined in Eq. (10). We then took  $\mathbf{x} = \mathbf{W}^\dagger \gamma_{\log}(\mathbf{q})$ , and drew  $\mathbf{y} \sim \mathbf{p}$  (here  $\mathbf{W}^\dagger$  denotes the pseudo-inverse of  $\mathbf{W}$ ). It can be verified that this gives  $\mathbf{q}(\mathbf{x}) = \mathbf{q} = \gamma_{\log}^{-1}(\mathbf{W}\mathbf{x})$ , and therefore, taking the function class  $\mathcal{F}$  in our algorithm to be the class of linear functions (i.e., functions of the form  $\mathbf{x} \mapsto \mathbf{V}\mathbf{x}$  for  $\mathbf{V} \in \mathbb{R}^{37 \times 100}$ ) suffices to learn a Bayes optimal multi-label classifier.

With the above settings, we used our algorithm (with logistic binary loss  $\phi_{\log}$  and linear function class) to learn a multi-label classifier from increasingly large training samples drawn according to the above distribution, and measured the  $F_1$  performance on a large test set of 15,000 data points drawn from the same distribution. The results are shown in Figure 1. As can be seen, our algorithm indeed converges to a Bayes optimal classifier for  $F_1$ .

## 7.2. Real Data: Comparison with Other Algorithms

In the second set of experiments, we evaluated the performance of our algorithm on various benchmark multi-label data sets drawn from the Mulan repository<sup>6</sup>. Details of the data sets are provided in Table 1. All the data sets come with prescribed train/test splits. After training our models on the training set, we measure the *instance-averaged*  $F_1$  performance on the test set (i.e., we compute the multi-label  $F_1$ -measure on each test example and take the average).

We compared with the following algorithms: EFP (Dembczynski et al., 2013), LIMO (label-wise version recommended for instance-averaged  $F_1$ ) (Wu & Zhou, 2017), and BR (which treats the  $s$  labels as conditionally independent and trains  $s$  binary logistic regression classifiers, one for each label). All algorithms were trained to learn linear models. Regularization parameters (for regularized logistic regression in our algorithm, EFP, and BR; and for the margin-based objective in LIMO) were chosen by 5-fold cross-validation on the training set from  $\{10^{-4}, \dots, 10^3\}$

<sup>6</sup><http://mulan.sourceforge.net/datasets-mlc.html>

Table 1. Multi-label data sets used in experiments in Section 7.2.

Data set	# train	# test	# labels	# features
Scene	1211	1196	6	294
Yeast	1500	917	14	103
Birds	322	323	19	260
Medical	333	645	45	1449
Enron	1123	579	53	1001
Mediamill	30993	12914	101	120

Table 2. Comparison of  $F_1$  performance of our algorithm with other MLC algorithms on various Mulan multi-label data sets. Higher values are better. See Section 7.2 for details and for an explanation of the asterisks for the Birds data set.

Data set	Our algorithm	EFP	LIMO	BR
Scene	<b>0.7445</b>	0.7426	0.6325	0.6009
Yeast	<b>0.6571</b>	0.6558	0.4914	0.6065
Birds	<b>*0.5836</b>	*0.5293	0.5463	0.5510
Medical	0.7557	<b>0.7685</b>	0.7237	0.6507
Enron	0.5868	<b>0.6204</b>	0.5764	0.5455
Mediamill	<b>0.5642</b>	0.5600	0.5135	0.5229

(for all algorithms, the parameter value maximizing average  $F_1$ -measure across the 5 folds was selected). For our algorithm and EFP, as discussed in Section 6, we generally implemented the algorithms to estimate only a small subset of the  $s^2 + 1$  statistics in  $\mathbf{q}(x)$  (only those corresponding to numbers of active labels seen in the training data); for the Birds data set, this resulted in poor performance for both algorithms, and so for this data set we trained both algorithms to perform a full estimation of all  $s^2 + 1$  statistics.

The results are shown in Table 2 (the asterisks in the results for the Birds data set denote the full estimation of  $s^2 + 1$  statistics for this data set, as discussed above). As expected, the performance of our algorithm is similar to that of EFP. BR, as expected, is generally a relatively weak baseline. LIMO is sometimes competitive, but since it aims to simultaneously optimize several multi-label performance measures, we do not expect it to outperform algorithms designed for a specific performance measure, and indeed this is borne out in our experiments.

## 8. Conclusion

We have provided a family of convex calibrated surrogate losses for the multi-label  $F_\beta$ -measure, together with a quantitative regret transfer bound. Our surrogates effectively decompose the  $F_\beta$  learning problem over  $s$  labels into (at most)  $s^2 + 1$  binary class probability estimation (CPE) problems. The regret transfer bound allows us to transfer any regret guarantees on the binary CPE learners to regret guarantees on the overall  $F_\beta$  learner. Although motivated from a different viewpoint, like the EFP algorithm of Dembczynski et al. (2013), our algorithm can also be viewed as a type of ‘plug-in’ algorithm for the  $F_\beta$ -measure. While we have described the algorithm in the context of multi-label classification, the algorithm can also be used for binary sequence labeling tasks where the  $F_\beta$ -measure is useful.



## Acknowledgments

This material is based upon work supported in part by the US National Science Foundation (NSF) under Grant Nos. 1934876 and 1717290 (awarded to SA). SA is also supported in part by the US National Institutes of Health (NIH) under Grant No. U01CA214411. HGR thanks the Robert Bosch Center for Data Science and Artificial Intelligence at IITM for its support. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation, the National Institutes of Health, or the Robert Bosch Center.

## References

- Agarwal, S. Surrogate regret bounds for bipartite ranking via strongly proper losses. *Journal of Machine Learning Research*, 15:1653–1674, 2014.
- Bartlett, P. L., Jordan, M., and McAuliffe, J. Convexity, classification and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.
- Dembczynski, K., Cheng, W., and Hüllermeier, E. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 279–286, 2010.
- Dembczynski, K., Waegeman, W., Cheng, W., and Hüllermeier, E. An exact algorithm for F-measure maximization. In *Advances in Neural Information Processing Systems 24*, pp. 1404–1412, 2011.
- Dembczynski, K., Jachnik, A., Kotlowski, W., Waegeman, W., and Hüllermeier, E. Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pp. 1130–1138, 2013.
- Duchi, J., Mackey, L., and Jordan, M. On the consistency of ranking algorithms. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.
- Gao, W. and Zhou, Z. On the consistency of multi-label learning. *Artificial Intelligence*, 199-200:22–44, 2013.
- Joachims, T. A support vector method for multivariate performance measures. In Raedt, L. D. and Wrobel, S. (eds.), *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pp. 377–384, 2005.
- Koyejo, O., Natarajan, N., Ravikumar, P., and Dhillon, I. S. Consistent multilabel classification. In *Advances in Neural Information Processing Systems 28*, pp. 3321–3329, 2015.
- Parambath, S. P., Usunier, N., and Grandvalet, Y. Optimizing F-measures by cost-sensitive classification. In *Advances in Neural Information Processing Systems 27*, pp. 2123–2131, 2014.
- Petterson, J. and Caetano, T. S. Reverse multi-label learning. In *Advances in Neural Information Processing Systems 23*, pp. 1912–1920, 2010.
- Petterson, J. and Caetano, T. S. Submodular multi-label learning. In *Advances in Neural Information Processing Systems 24*, pp. 1512–1520, 2011.
- Pillai, I., Fumera, G., and Roli, F. Designing multi-label classifiers that maximize F measures: State of the art. *Pattern Recognition*, 61:394–404, 2017.
- Ramaswamy, H. G. *Design and Analysis of Consistent Algorithms for Multiclass Learning Problems*. PhD thesis, Indian Institute of Science, 2015.
- Ramaswamy, H. G., Agarwal, S., and Tewari, A. Convex calibrated surrogates for low-rank loss matrices with applications to subset ranking losses. In *Advances in Neural Information Processing Systems*, 2013.
- Ramaswamy, H. G., Babu, B. S., Agarwal, S., and Williamson, R. C. On the consistency of output code based learning algorithms for multiclass learning problems. In *Proceedings of the 27th Conference on Learning Theory (COLT)*, pp. 885–902, 2014.
- Ramaswamy, H. G., Tewari, A., and Agarwal, S. Convex calibrated surrogates for hierarchical classification. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 1852–1860, 2015.
- Reid, M. D. and Williamson, R. C. Composite binary losses. *Journal of Machine Learning Research*, 11:2387–2422, 2010.
- Steinwart, I. How to compare different loss functions and their risks. *Constructive Approximation*, 26:225–287, 2007.
- Tewari, A. and Bartlett, P. L. On the consistency of multi-class classification methods. *Journal of Machine Learning Research*, 8:1007–1025, 2007.
- Tsochantiridis, I., Joachims, T., Hoffman, T., and Altun, Y. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.
- Wu, X. and Zhou, Z. A unified view of multi-label performance measures. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 3780–3788, 2017.

- Ye, N., Chai, K. M. A., Lee, W. S., and Chieu, H. L. Optimizing F-measure: A tale of two approaches. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.
- Zhang, M. and Zhou, Z. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014.
- 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.

# Convex Calibrated Surrogates for the Multi-Label F-Measure

## Supplementary Material

### Implementation of ‘decode’

In order to solve the combinatorial optimization problem involved in the mapping  $\text{decode} : \mathbb{R}^{s^2+1} \rightarrow \{0, 1\}^s$  as defined in Eq. (7) efficiently, we make use of an  $O(s^3)$ -time procedure due to Dembczynski et al. (2011). Specifically, Dembczynski et al. (2011) gave a procedure that, given a certain set of  $s^2 + 1$  statistics of the true conditional distribution  $p(\mathbf{y}|x)$  at a point  $x \in \mathcal{X}$ , computes in  $O(s^3)$  time a Bayes optimal multi-label prediction  $h^*(x) \in \{0, 1\}^s$  at that point with respect to the  $F_1$ -measure by solving a similar combinatorial optimization problem (the approach generalizes easily to the  $F_\beta$ -measure for general  $\beta$ ). Our algorithm (Algorithm 2) can be viewed as effectively estimating the same  $s^2 + 1$  statistics from the training sample  $S$ ; in particular, once a scoring function  $\mathbf{f}_S : \mathcal{X} \rightarrow \mathbb{R}^{s^2+1}$  is learned by minimizing our surrogate loss  $\psi$ , the estimated statistics at a point  $x \in \mathcal{X}$  are given by  $\gamma^{-1}(\mathbf{f}_S(x))$  (where  $\gamma^{-1}$  is the inverse of the link function  $\gamma : [0, 1] \rightarrow \mathbb{R}$  associated with the strictly proper composite binary loss  $\phi$  used in our surrogate, and is applied element-wise to  $\mathbf{f}_S(x)$ ). Our ‘decode’ mapping effectively corresponds to estimating a Bayes optimal prediction at  $x$  using these estimated statistics; we can therefore apply the procedure of Dembczynski et al. (2011) to these estimated statistics.

The implementation below is described for a general input vector  $\mathbf{u} \in \mathbb{R}^{s^2+1}$  (see Eq. (7)); in our  $F_\beta$  learning algorithm, to make a prediction at  $x \in \mathcal{X}$ , it would be applied to  $\mathbf{u} = \mathbf{f}_S(x)$ . The overall idea is that the combinatorial search over  $\hat{\mathbf{y}} \in \{0, 1\}^s$  is stratified over the  $s + 1$  sets  $\hat{\mathcal{Y}}_l = \{\hat{\mathbf{y}} \in \{0, 1\}^s : \|\hat{\mathbf{y}}\|_1 = l\}$ ,  $l \in \{0, 1, \dots, s\}$ ; to find an optimal element  $\hat{\mathbf{y}}^{l,*}$  within each set  $\hat{\mathcal{Y}}_l$ , one need only solve a problem of the form  $\hat{\mathbf{y}}^{l,*} \in \operatorname{argmin}_{\hat{\mathbf{y}} \in \hat{\mathcal{Y}}_l} \sum_{j=1}^s \hat{y}_j T_{jl}$  for certain numbers  $T_{jl}$ , which can be done simply by finding the smallest  $l$  numbers among  $\{T_{jl} : j \in [s]\}$  and setting the corresponding  $l$  entries of  $\hat{\mathbf{y}}^{l,*}$  to 1 (and remaining entries to 0). Solving these  $s + 1$  subproblems and picking the best solution among them takes a total of  $O(s^2 \ln(s))$  time; computing the  $s^2$  numbers  $T_{jl}$  involves a matrix multiplication that takes a total of  $O(s^3)$  time.<sup>7</sup>

---

#### Algorithm 2 Decode

---

- 1: **Input:** Vector  $\mathbf{u} = (u_0, (u_{jk})_{j,k=1}^s)^\top \in \mathbb{R}^{s^2+1}$
- 2: **Parameters:** Link function  $\gamma : [0, 1] \rightarrow \mathbb{R}$
- 3: Define matrices  $\mathbf{Q} \in [0, 1]^{s \times s}$  and  $\mathbf{V} \in \mathbb{R}^{s \times s}$  as follows:

$$\begin{aligned} Q_{jk} &= \gamma^{-1}(u_{jk}) \\ V_{kl} &= \frac{-(1 + \beta)^2}{\beta^2 k + l} \end{aligned}$$

- 4: Compute  $\mathbf{T} = \mathbf{QV}$  // matrix multiplication,  $O(s^3)$  time
- 5: **For**  $l = 1 \dots s$ : // for loop takes total  $O(s^2 \ln(s))$  time
- 6: Find the  $l$  smallest numbers among  $\{T_{jl} : j \in [s]\}$ ; call the corresponding indices  $j_1^l, \dots, j_l^l$
- 7: Define  $\hat{\mathbf{y}}^{l,*} \in \{0, 1\}^s$  as follows:

$$\hat{y}_j^{l,*} = \begin{cases} 1 & \text{if } j \in \{j_1^l, \dots, j_l^l\} \\ 0 & \text{otherwise.} \end{cases} \quad // \text{ this solves } \hat{\mathbf{y}}^{l,*} \in \operatorname{argmin}_{\hat{\mathbf{y}} \in \hat{\mathcal{Y}}_l} \sum_{j=1}^s \hat{y}_j T_{jl}$$

- 8: Set  $z_l^* = \sum_{j=1}^s \hat{y}_j^{l,*} T_{jl}$
- 9: **End for**
- 10: Pick  $\hat{\mathbf{y}}^* \in \{0, 1\}^s$  as follows:

$$\hat{\mathbf{y}}^* \in \operatorname{argmin}_{\hat{\mathbf{y}} \in \{\mathbf{0}, \hat{\mathbf{y}}^{1,*}, \dots, \hat{\mathbf{y}}^{s,*}\}} - \mathbf{1}(\hat{\mathbf{y}} = \mathbf{0}) \cdot \gamma^{-1}(u_0) + \mathbf{1}(\hat{\mathbf{y}} \neq \mathbf{0}) \cdot z_{\|\hat{\mathbf{y}}\|_1}^*$$

- 11: **Output:**  $\hat{\mathbf{y}}^* \in \{0, 1\}^s$
- 

<sup>7</sup>One could in principle use faster matrix multiplication methods that take  $o(s^3)$  time, but in practice, this would be helpful for only extremely large values of  $s$ .