A Unified View of Multi-Label Performance Measures

Xi-Zhu Wu [1] **Zhi-Hua Zhou** [1]

Abstract

Multi-label classification deals with the problem where each instance is associated with multiple class labels. Because evaluation in multi-label classification is more complicated than single-label setting, a number of performance measures have been proposed. It is noticed that an algorithm usually performs differently on different measures. Therefore, it is important to understand which algorithms perform well on which measure(s) and why. In this paper, we propose a unified margin view to revisit eleven performance measures in multi-label classification. In particular, we define *label-wise* margin and *instance-wise* margin, and prove that through maximizing these margins, different corresponding performance measures are to be optimized. Based on the defined margins, a max-margin approach called LIMO is designed and empirical results validate our theoretical findings.

1. Introduction

Multi-label classification aims to build classification models for objects assigned with *multiple* labels simultaneously, which is a common learning paradigm in real-world applications. In text categorization, a document may be associated with a range of topics, such as *science*, *entertainment*, and *news* (Schapire & Singer, 2000); in image classification, an image can have both *field* and *mountain* tags (Boutell et al., 2004); in music information retrieval, a piece of music can convey various messages such as *classic*, *piano* and *passionate* (Turnbull et al., 2008).

In traditional supervised classification, generalization performance of the learning system is usually evaluated by accuracy, or F-measure if misclassification costs are unequal. In contrast to single-label classification, performance evaluation in multi-label classification is more complicated, as each instance can be associated with multiple labels simultaneously. For example, it is difficult to tell which mistake of the following two cases is more serious: one instance with three incorrect labels vs. three instances each with one incorrect label. Therefore, a number of performance measures focusing on different aspects have been proposed, such as *Hamming loss*, *ranking loss*, *one-error*, *average precision*, *coverage* (Schapire & Singer, 2000), *micro-F1* and *macro-F1* (Tsoumakas et al., 2011).

Multi-label learning algorithms usually perform differently on different measures; however, there are only a few studies about multi-label performance measures. Dembczynski et al. (2010) showed that Hamming loss and subset 0/1 loss could not be optimized at the same time. Gao & Zhou (2013) proposed to study the Bayes consistency of surrogate losses for multi-label learning; they proved that none of convex surrogate loss is consistent with ranking loss, and gave a consistent surrogate loss function for Hamming loss in deterministic case. There are a number of studies about F-measure, mostly focusing on single-label tasks, including multi-label learning as application. For example, Ye et al. (2012) gave justifications and connections about F-measure optimization using decision theoretic approaches (DTA) and empirical utility maximization approaches (EUM). Later, Waegeman et al. (2014) studied the F-measure optimality of inference algorithms from the DTA perspective. Koyejo et al. (2015) devoted to study of EUM optimal multi-label classifiers. These theoretical studies offer much insight, though lacking a unified understanding of relation among a variety of multi-label performance measures. Moreover, some performance measures which have been popularly used in evaluation (Zhang & Wu, 2015) have not been theoretically studied.

In this paper, we try to disclose some shared properties among different measures and establish a unified understanding for multi-label performance evaluation. We propose a *margin* view to revisit eleven commonly used multi-label performance measures, including Hamming loss, ranking loss, one-error, coverage, average precision, macro-, micro- and instance-averaging F-measures and AUCs. Specifically, we propose the concepts of *label-wise* margin and *instance-wise* margin, based on which the corresponding *effectiveness* of multi-label classifiers is defined and then used as bridge to connect different perfor-

[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China. Correspondence to: Zhi-Hua Zhou <zhouzh@lamda.nju.edu.cn>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

mance measures. Our theoretical results show that by maximizing instance-wise margin, macro-AUC, macro-F1 and Hamming loss are to be optimized, whereas by maximizing label-wise margin, the other eight performance measures except micro-AUC are to be optimized. Inspired by the theoretical findings, we design the LIMO (Label-wise and Instance-wise Margins Optimization) approach to maximize both the two margins. Experiments validate our theoretical findings and demonstrate a flexible way to optimize different measures through one approach by different parameter settings.

The rest of the paper is organized as follows. Section 2 introduces the notation and definitions of eleven multi-label performance measures. Section 3 proposes the label-wise and instance-wise margins, and presents our theoretical results. Section 4 presents the LIMO approach. Section 5 reports the results of experiments. Finally, Section 6 concludes and indicates several future issues.

2. Preliminaries

2.1. Notation

Assume that $\boldsymbol{x}_i \in \mathbb{R}^{d\times 1}$ is a real value instance vector, $\boldsymbol{y}_i \in \{0,1\}^{l\times 1}$ is a label vector for $\boldsymbol{x}_i$. m denotes the number of training samples. Therefore y_{ij} ($i \in \{1,\ldots,m\}, j \in \{1,\ldots,l\}$) means the jth label of the ith instance, and $y_{ij} = 1$ or 0 means the jth label is relevant or irrelevant. The instance matrix is $\boldsymbol{X} \in \mathbb{R}^{m\times d}$ and the label matrix is $\boldsymbol{Y} \in \{0,1\}^{m\times l}$. $H : \mathbb{R}^d \to \{0,1\}^l$ is the multi-label classifier, which consists of l models, one for a label, so $H = \{h_1,\ldots,h_l\}$ and $h_j(\boldsymbol{x}_i)$ denotes the prediction of y_{ij}. Moreover, $F : \mathbb{R}^d \to \mathbb{R}^l$ is the multi-label predictor and the predicted value can be regarded as the confidence of relevance. Similarly, F can be decomposed as $\{f_1,\ldots,f_l\}$ where $f_j(\boldsymbol{x}_i)$ denotes the predicted value of y_{ij}.

H can be induced from F via thresholding functions. For example, $h_j(\boldsymbol{x}_i) = [\![f_j(\boldsymbol{x}_i) > t(\boldsymbol{x}_i)]\!]$ uses a thresholding function based on the instance $\boldsymbol{x}_i$ and outputs 1 if predicted value is higher than the threshold. $[\![\pi]\!]$ returns 1 if predicate π holds, and 0 otherwise.

For simplification, we use $\boldsymbol{Y}_{i\cdot}$ to denote the ith row vector and $\boldsymbol{Y}_{\cdot j}$ to denote the jth column vector of the label matrix. Furthermore, $Y_{i\cdot}^+$ (or $Y_{i\cdot}^-$) denotes the index set of relevant (or irrelevant) labels of $\boldsymbol{Y}_{i\cdot}$. Formally, $Y_{i\cdot}^+ = \{j \,|y_{ij} = 1\}$ and $Y_{i\cdot}^- = \{j \,|y_{ij} = 0\}$. In terms of jth column of label matrix, $Y_{\cdot j}^+ = \{i \,|y_{ij} = 1\}$ denotes the index set of positive instances of the jth label and $Y_{\cdot j}^- = \{i \,|y_{ij} = 0\}$ denotes the set of negative instances similarly. We use $|\cdot|$ to denote the cardinality of a set, thus, the number of relevant labels of $\boldsymbol{x}_i$ is $|Y_{i\cdot}^+|$.

2.2. Multi-label Performance Measures

Table 1 summarizes the eleven multi-label performance measures commonly used in previous studies. The first five measures (Hamming loss, ranking loss, one-error, coverage, average precision) are considered in Schapire & Singer (2000) and a multitude of works, e.g., Huang et al. (2012) and Zhang & Wu (2015). The next six measures are extensions of F-measure and AUC (the Area Under the ROC Curve) in multi-label classification via different averaging strategies. These F-measures are popluar both in algorithm evaluation (Liu & Tsang, 2015) and theoretical analysis (Koyejo et al., 2015). AUCs are used for algorithm evaluation such as in Lampert (2011), Pham et al. (2015) and Zhang & Wu (2015).

Some of these measures are defined on classifier H, and they care about the binary classification performance. While some of these measures are defined on predictor F, and they usually measure the ranking performance of the predictor. We have noticed that some performance measures on ranking are ill-defined when F is a constant function. For example, if F outputs 1 for all labels, then *one-error*$(F) = 0$, *coverage*$(F) = 0$ and various AUCs will be 1, which are the optimal values respectively. In multi-label learning community, there is often an underlying assumption that a total ranking can be induced from continuous real-value predictions, which is common in practical cases. In this paper, we still stick to the convention in previous works and assume that no tie happens in continuous prediction to solve this definition flaw.

3. Theoretical Results

Here we define two new concepts: label-wise margin and instance-wise margin.

Definition 1. *Given a multi-label predictor $F : \mathbb{R}^d \to \mathbb{R}^l$ and $F = \{f_1,\ldots,f_l\}$, a training set $(\boldsymbol{X},\boldsymbol{Y})$, the **label-wise margin** on instance $\boldsymbol{x}_i$ is defined as:*

$$\gamma_i^{label} = \min_{u,v}\{f_u(\boldsymbol{x}_i) - f_v(\boldsymbol{x}_i) \mid (u,v) \in Y_{i\cdot}^+ \times Y_{i\cdot}^-\}.$$

$Y_{i\cdot}^+ \times Y_{i\cdot}^-$ is the set of all the (relevant, irrelevant) label index pairs of instance i.

Definition 2. *Given a multi-label predictor $F : \mathbb{R}^d \to \mathbb{R}^l$ and $F = \{f_1,\ldots,f_l\}$, a training set $(\boldsymbol{X},\boldsymbol{Y})$, the **instance-wise margin** on label $\boldsymbol{Y}_{\cdot j}$ is defined as:*

$$\gamma_j^{inst} = \min_{a,b}\{f_j(\boldsymbol{x}_a) - f_j(\boldsymbol{x}_b) \mid (a,b) \in Y_{\cdot j}^+ \times Y_{\cdot j}^-\}.$$

$Y_{\cdot j}^+ \times Y_{\cdot j}^-$ is the set of all the (positive, negative) instance index pairs of label j.

Label-wise margin and instance-wise margin describe the discriminative ability of F. The larger the label-wise mar-

Table 1. Definitions of eleven multi-label performance measures

Measure	Formulation	Note
Hamming loss	$hloss(H) = \frac{1}{ml}\sum_{i=1}^{m}\sum_{j=1}^{l}[\![h_{ij} \neq y_{ij}]\!]$	The fraction of misclassified labels
ranking loss	$rloss(F) = \frac{1}{m}\sum_{i=1}^{m}\frac{\lvert\mathcal{S}^i_{\text{rank}}\rvert}{\lvert Y^+_{i\cdot}\rvert\lvert Y^-_{i\cdot}\rvert}$ $\mathcal{S}^i_{\text{rank}} = \{(u,v) \lvert f_u(\boldsymbol{x}_i) \leq f_v(\boldsymbol{x}_i), (u,v) \in Y^+_{i\cdot} \times Y^-_{i\cdot}\}$	The average fraction of reversely ordered label pairs of each instance.
one-error	$one\text{-}error(F) = \frac{1}{m}\sum_{i=1}^{m}[\![\arg\max F(\boldsymbol{x}_i) \notin Y^+_{i\cdot}]\!]$	The fraction of instances whose most confident label is irrelevant.
coverage	$coverage(F) = \frac{1}{m}\sum_{i=1}^{m}[\![\max_{j\in Y^+_{i\cdot}} rank_F(\boldsymbol{x}_i, j) - 1]\!]$	The number of more labels on average should include to cover all relevant labels
average precision	$avgprec(F) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{\lvert Y^+_{i\cdot}\rvert}\sum_{j\in Y^+_{i\cdot}}\frac{\lvert\mathcal{S}^{ij}_{\text{precision}}\rvert}{rank_F(\boldsymbol{x}_i, j)}$ $\mathcal{S}^{ij}_{\text{precision}} = \{k \in Y^+_{i\cdot} \lvert rank_F(\boldsymbol{x}_i, k) \leq rank_F(\boldsymbol{x}_i, j)\}$	The average fraction of relevant labels ranked higher than one other relevant label.
macro-F1	$\textit{macro-F1}(H) = \frac{1}{l}\sum_{j=1}^{l}\frac{2\sum_{i=1}^{m} y_{ij}h_{ij}}{\sum_{i=1}^{m} y_{ij} + \sum_{i=1}^{m} h_{ij}}$	F-measure averaging on each label.
instance-F1	$\textit{instance-F1}(H) = \frac{1}{m}\sum_{i=1}^{m}\frac{2\sum_{j=1}^{l} y_{ij}h_{ij}}{\sum_{j=1}^{l} y_{ij} + \sum_{j=1}^{l} h_{ij}}$	F-measure averaging on each instance.
micro-F1	$\textit{micro-F1}(H) = \frac{2\sum_{j=1}^{l}\sum_{i=1}^{m} y_{ij}h_{ij}}{\sum_{j=1}^{l}\sum_{i=1}^{m} y_{ij} + \sum_{j=1}^{l}\sum_{i=1}^{m} h_{ij}}$	F-measure averaging on the prediction matrix.
macro-AUC	$\textit{macro-AUC}(F) = \frac{1}{l}\sum_{j=1}^{l}\frac{\lvert\mathcal{S}^j_{\text{macro}}\rvert}{\lvert Y^+_{\cdot j}\rvert\lvert Y^-_{\cdot j}\rvert}$ $\mathcal{S}^j_{\text{macro}} = \{(a,b) \in Y^+_{\cdot j} \times Y^-_{\cdot j} \lvert f_j(\boldsymbol{x}_a) \geq f_j(\boldsymbol{x}_b)\}$	AUC averaging on each label. $\mathcal{S}_{\text{macro}}$ is the set of correctly ordered instance pairs on each label.
instance-AUC	$\textit{instance-AUC}(F) = \frac{1}{m}\sum_{i=1}^{m}\frac{\lvert\mathcal{S}^i_{\text{instance}}\rvert}{\lvert Y^+_{i\cdot}\rvert\lvert Y^-_{i\cdot}\rvert}$ $\mathcal{S}^i_{\text{instance}} = \{(u,v) \in Y^+_{i\cdot} \times Y^-_{i\cdot} \lvert f_u(\boldsymbol{x}_i) \geq f_v(\boldsymbol{x}_i)\}$	AUC averaging on each instance. $\mathcal{S}_{\text{instance}}$ is the set of correctly ordered label pairs on each instance.
micro-AUC	$\textit{micro-AUC}(F) = \frac{\lvert\mathcal{S}_{\text{micro}}\rvert}{(\sum_{i=1}^{m}\lvert Y^+_{i\cdot}\rvert)\cdot(\sum_{i=1}^{m}\lvert Y^-_{i\cdot}\rvert)}$ $\mathcal{S}_{\text{micro}} = \{(a,b,i,j) \lvert (a,b) \in Y^+_{\cdot i} \times Y^-_{\cdot j},\ f_i(\boldsymbol{x}_a) \geq f_j(\boldsymbol{x}_b)\}$	AUC averaging on prediction matrix. $\mathcal{S}_{\text{micro}}$ is the set of correct quadruples.

gin, the easier to distinguish relevant and irrelevant labels of an instance. Meanwhile, the larger the instance-wise margin, the easier for F to distinguish positive and negative instances of a particular label. Therefore, we want to maximize label-wise/instance-wise margin to get better performance.

Although we prefer maximizing these two margins, with respect to performance measures, the objective can be relaxed. We define three properties a predictor F can have: label-wise effective, instance-wise effective and double effective.

Definition 3. *If all the label-wise margins of F on a dataset $D = (\boldsymbol{X}, \boldsymbol{Y})$ are positive, this predictor F is **label-wise effective** on D.*

Definition 4. *If all the instance-wise margins of F on a dataset $D = (\boldsymbol{X}, \boldsymbol{Y})$ are positive, this predictor F is **instance-wise effective** on D.*

Definition 5. *If all the label-wise margins **and** instance-wise margins of F on a dataset $D = (\boldsymbol{X}, \boldsymbol{Y})$ are positive, this predictor F is **double effective** on D.*

Roughly speaking, label-wise effective means F can exactly distinguish relevant and irrelevant labels of each instance and instance-wise effective means F can exactly distinguish positive and negative instances of every label. Not surprisingly, double effective F has the strongest ability in distinguishing.

In the next two subsections, we use the effectiveness to an-

alyze different performance measures, and summarize the analysis results in Section 3.3.

3.1. Performance Measures on Ranking

Several multi-label performance measures can be empirically optimized according to the following theorems:

Theorem 1. *If a multi-label predictor F is label-wise effective on D, then ranking loss, one-error, coverage, average precision and instance-AUC are optimized on the dataset.*

Proof. (a) Ranking loss: From the definition of label-wise effective, for every pair $(u, v) \in Y_{i\cdot}^+ \times Y_{i\cdot}^-$, we have $f_u(\boldsymbol{x}_i) > f_v(\boldsymbol{x}_i)$. Therefore, the reversed set $\mathcal{S}_{\text{rank}}^i$ (in Table 1 ranking loss) is empty and the cardinality of the set is zero, which implies the cardinality sum of all reversed sets $rloss(F) = 0$. Ranking loss is optimized.

(b) One-error: For a label-wise effective F, because label-wise margin is positive on an instance $\boldsymbol{x}_i$, we have:

$$\max_u f_u(\boldsymbol{x}_i) > \max_v f_v(\boldsymbol{x}_i), \forall u \in Y_{i\cdot}^+, \forall v \in Y_{i\cdot}^-.$$

Then

$$\forall \boldsymbol{x}_i,\ \arg\max F(\boldsymbol{x}_i) \in Y_{i\cdot}^+.$$

Thus, $[\![\arg\max F(\boldsymbol{x}_i) \notin Y_{i\cdot}^+]\!] = 0$ for every instance $\boldsymbol{x}_i$, and $one\text{-}error(F) = 0$. One-error is optimized.

(c) Coverage: When F is label-wise effective, the maximum rank of a relevant label is less than the minimum rank of an irrelevant label, which means:

$$\begin{aligned} &\max_{u \in Y_{i\cdot}^+} rank_F(\boldsymbol{x}_i, u) < \min_{v \in Y_{i\cdot}^-} rank_F(\boldsymbol{x}_i, v), \\ &\max_{u \in Y_{i\cdot}^+} rank_F(\boldsymbol{x}_i, u) = |Y_{i\cdot}^+|. \end{aligned} \tag{1}$$

Therefore, coverage can be calculated as:

$$coverage(F) = \frac{1}{m} \sum\nolimits_{i=1}^{m} [|Y_{i\cdot}^+| - 1].$$

Which is the optimal value of coverage.

(d) Average precision: Assume that j is a relevant label of instance i, it follows from Equation (1) that:

$$rank_F(\boldsymbol{x}_i, j) = |\{k \in Y_{i\cdot}^+ | rank_F(\boldsymbol{x}_i, k) \leq rank_F(\boldsymbol{x}_i, j)\}|$$

Since $rank_F(\boldsymbol{x}_i, j)$ is exactly the definition of $\mathcal{S}_{\text{precision}}^{ij}$, $avgprec(F) = 1$, i.e, average precision is optimized.

(e) Instance-AUC: Because of label-wise effective, for an instance $\boldsymbol{x}_i$, we have:

$$f_u(\boldsymbol{x}_i) > f_v(\boldsymbol{x}_i), \forall (u, v) \in Y_{i\cdot}^+ \times Y_{i\cdot}^-.$$

Therefore, the size of the correct ordered prediction value pair on instance i is:

$$|\{(u, v) \in Y_{i\cdot}^+ \times Y_{i\cdot}^- | f_u(\boldsymbol{x}_i) \geq f_v(\boldsymbol{x}_i)\}| = |Y_{i\cdot}^+||Y_{i\cdot}^-|.$$

So *instance-AUC*$(F) = 1$ and instance-AUC is optimized.

□

Similar to the proof of instance-AUC, we can prove the result of macro-AUC:

Theorem 2. *If a multi-label predictor F is instance-wise effective on D, then macro-AUC is optimized.*

Proof. Because of instance-wise effective, for a label vector $\boldsymbol{Y}_{\cdot j}$, we have:

$$f_j(\boldsymbol{x}_a) > f_j(\boldsymbol{x}_b), \forall (a, b) \in Y_{\cdot j}^+ \times Y_{\cdot j}^-.$$

Therefore, the size of the correct ordered prediction value pair on label j is:

$$\{(a, b) \in Y_{\cdot j}^+ \times Y_{\cdot j}^- | f_j(\boldsymbol{x}_a) \geq f_j(\boldsymbol{x}_b)\} = |Y_{\cdot j}^+||Y_{\cdot j}^-|.$$

So *macro-AUC*$(F) = 1$ and macro-AUC is optimized. □

Micro-AUC sees the label matrix as a whole and cannot be optimized by instance-wise effective F or label-wise effective F. However, the double effective F is much more powerful. We now prove the following result of micro-AUC.

Theorem 3. *If a multi-label predictor F is double effective on D, then as the number of instances grows, micro-AUC is optimized.*

Proof. We first prove a result of random variables A_i, B, C. If n random variables $A_1, A_2, \cdots, A_n$ are drawn from uniform distribution $U(0, 1)$, for a random constant a, the event that at least one A_i is smaller than a is:

$$\Pr[\exists A_i,\ A_i \leq a] = 1 - (1 - a)^n.$$

Another random variable B is uniformly distributed in $(0, \min\{A_i\})$, and the probability that a random variable $C \sim U(0, 1)$ is bigger than B is:

$$\begin{aligned} \Pr[C > B] \geq &\Pr[(C \geq a) \wedge (\exists A_i,\ A_i \leq a)] \\ = &(1 - \frac{a}{2})[1 - (1 - a)^n]. \end{aligned} \tag{2}$$

For any small a, we can choose a large enough n to make Equation (2) close to 1.

Given a label matrix $\boldsymbol{Y} \in \{0, 1\}^{m \times l}$ and the corresponding prediction matrix $\boldsymbol{F} \in (0, 1)^{m \times l}$, because predictor F is double effective, the prediction matrix satisfies the following conditions:

$$\begin{aligned} &F_{ij} > F_{iu} \text{ if } Y_{ij} = 1 \wedge Y_{iu} = 0, \\ &F_{ij} > F_{vj} \text{ if } Y_{ij} = 1 \wedge Y_{vj} = 0. \end{aligned}$$

To force the value in $\boldsymbol{F}$ is in $(0, 1)$, we further assume a uniform distribution $F_{ij} \sim U(0, 1)$ when $Y_{ij} = 1$.

If $Y_{ij} = 0$, then F_{ij} should be less than F_{iu} if $Y_{iu} = 1$ and F_{vj} if $Y_{vj} = 1$. Suppose that the minimum value b is defined as:

$$b = \min\Big\{\min_{v}\{F_{vj}|Y_{vj} = 1\}, \min_{u}\{F_{iu}|Y_{iu} = 1\}\Big\}.$$

Then F_{ij} is drawn from $U(0, b)$. And we can choose a small constant value $a > b$.

According to Equation (2), the probability that a random pair (i, j, u, v) to be a correct micro pair is:

$$\begin{aligned} P_{\text{micro}} &= \Pr[F_{ij} > F_{uv}|Y_{ij} = 1, Y_{uv} = 0] \\ &\geq (1 - \frac{a}{2})[1 - (1 - a)^n], \\ \text{where } n &= \frac{k}{ml}(m + l - 2) \end{aligned}$$

In the practical case, the number of labels is proportional to the number of instances: $k \propto m$. We assume $k = pm$ where p is a constant smaller than l.

$$\lim_{m\to\infty} n = \lim_{m\to\infty} \frac{p}{l}(m + l - 2) = \infty,$$
$$\lim_{m\to\infty} \frac{|\mathcal{S}_{\text{micro}}|}{|(\sum_{i=1}^{m} |Y_{i\cdot}^{+}|) \cdot (\sum_{i=1}^{m} |Y_{i\cdot}^{-}|)|} = \lim_{m\to\infty} P_{\text{micro}} = 1.$$

Therefore, micro-AUC is to be optimized as the number of instances grows. □

With the above analysis, we can conclude that a label-wise effective F can optimize ranking loss, one-error, coverage, average precision, instance-AUC, micro-AUC and an instance-wise effective F can optimize macro-AUC. For micro-AUC, a double effective F can optimize it as the number of instances increases.

3.2. Performance Measures on Classification

As mentioned in Section 2.2, there are some measures evaluating classifier H instead of predictor F. There are many thresholding or binarization strategies (Fan & Lin, 2007; Fürnkranz et al., 2008; Read et al., 2011). For simplicity, we focus on two main strategies: thresholding on each instance and thresholding on each label.

A label-wise effective F can be equipped with a thresholding function based on each instance such as $t(\boldsymbol{x}_i)$ and construct the H by $h_j(\boldsymbol{x}_i) = [\![f_j(\boldsymbol{x}_i) > t(\boldsymbol{x}_i)]\!]$. However, using $t(\boldsymbol{x}_i)$ on an instance-wise effective F is unreasonable since the predicted values on different labels may not be comparable. In a word, we should use suitable threshold function on different effective Fs, i.e., $t(\boldsymbol{x}_i)$ on each instance for label-wise effective F, and t_j on each label for instance-wise effective F. It is reasonable to use either $t(\boldsymbol{x}_i)$ or t_j for double effective F.

To formally analyze the performance measures on classification, we define the threshold error:

Definition 6. *Given a descending ordered real-value sequence $x_1, x_2, \ldots, x_k$ with an optimal cut number c^*, where $c^* \in \mathbb{N}$ and $1 \leq c^* \leq k$. For a real value threshold $t \in (x_k - 1, x_1 + 1)$, the **threshold error** $\epsilon = |\arg\min_i(x_i) - c^*|$ where $x_i > t$.*

Intuitively, the threshold error ϵ counts how many items are incorrectly classified on a descending ordered sequence where the correct answer is c^*. Based on the threshold error, we propose the following theorems about performance measures on classification.

Theorem 4. *For a label-wise effective F, if the thresholding function makes at most ϵ_i error on each instance i, the micro-F1, instance-F1 and Hamming loss are bounded as follows:*

$$\begin{aligned} \textit{micro-F1}(H) &= \textit{instance-F1}(H) \\ &\geq \frac{1}{m}\sum_{i=1}^{m} \min\Big\{\frac{2(|Y_{i\cdot}^{+}| - \epsilon_i)}{2|Y_{i\cdot}^{+}| - \epsilon_i}, \frac{2|Y_{i\cdot}^{+}|}{2|Y_{i\cdot}^{+}| + \epsilon_i}\Big\}, \\ \textit{hloss}(H) &\leq \frac{1}{ml}\sum\nolimits_{i=1}^{m} \epsilon_i. \end{aligned}$$

The main idea of the above theorem is that, given the threshold error and the number of relevant labels, we can compute the gap between the worst possible and the perfect contingency table. Hence the F-measure is based on the contingency table, the lower bound can be deduced. The detailed proof of Theorem 4 is in Appendix A.1.

Similar to Theorem 4, we can prove the results for label-wise effective F:

Theorem 5. *For an instance-wise effective F, if the thresholding function makes at most ϵ_j error on each label j, then the macro-F1 and Hamming loss are bounded as follows:*

$$\begin{aligned} \textit{macro-F1}(H) &\geq \frac{1}{l}\sum_{j=1}^{l} \min\Big\{\frac{2(|Y_{\cdot j}^{+}| - \epsilon_j)}{2|Y_{\cdot j}^{+}| - \epsilon_j}, \frac{2|Y_{\cdot j}^{+}|}{2|Y_{\cdot j}^{+}| + \epsilon_j}\Big\}, \\ \textit{hloss}(H) &\leq \frac{1}{ml}\sum\nolimits_{j=1}^{l} \epsilon_j. \end{aligned}$$

The detailed proof of Theorem 5 is in Appendix A.2.

With the above analysis, we can conclude that a label-wise effective F can optimize instance-F1 and micro-F1, an instance-wise effective F can optimize macro-F1. Both the two effective Fs can optimize Hamming loss. For a double effective F, because it enjoys both the properties, it can optimize all the above mentioned performance measures if proper thresholds are used.

Table 2. Summary of performance measures optimized by x-effective multi-label predictor (F). '✓' means F in this cell is proved to optimize this measure; '✗' means F in this cell does not necessarily optimize the measure; '•'/'○' means the calculation is with/without thresholding.

Measure	x-effective F			Threshold
	label-wise	inst-wise	double	
ranking loss	✓	✗	✓	○
avg. precision	✓	✗	✓	○
one-error	✓	✗	✓	○
coverage	✓	✗	✓	○
instance-AUC	✓	✗	✓	○
macro-AUC	✗	✓	✓	○
micro-AUC	✗	✗	✓	○
macro-F1	✗	✓	✓	•
instance-F1	✓	✗	✓	•
micro-F1	✓	✗	✓	•
Hamming loss	✓	✓	✓	•

3.3. Summary

Table 2 summarizes our theoretical results in Section 3.1 and 3.2. Each row shows the results of one multi-label performance measure. Note that double effective is a special case of label-wise effective and instance-wise effective and thus, if one performance measure is optimized by either label-wise or instance-wise effective predictor, it will also be optimized by double effective predictor.

In the light of the analysis, the performance on different performance measures through optimizing margins can be expected. For example, if one maximizes instance-wise margin on each label, s/he will get good performance on macro-AUC but may suffer higher loss on ranking loss, coverage and some other measures where '✗' marked in the inst-wise column. If one tries to maximize the label-wise margin but pay no attention to instance-wise margin, s/he may perform well on average precision but poor on macro-F1 (e.g., Elisseeff & Weston (2002)). Maximzing both the label-wise margin and instance-wise margins to get a double effective F is expected to be the best choice.

4. The LIMO Approach

The above analysis reveals that maximizing different margins will optimize different measures, and if possible, double effective F is prefered since it enjoys the benefits of maximizing both the label-wise margin and the instance-wise margin. Therefore, we propose the LIMO approach. LIMO is a single approach which can optimize both the two margins, and it can also be degenerated to optimize either margin seperately via parameter setting.

4.1. Formulation

Suppose that F is a linear predictor, which means $F(\boldsymbol{X}) = \boldsymbol{W}^T\boldsymbol{X}$ where $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_l]$. We propose the following formulation:

$$\begin{aligned}\arg\min_{\boldsymbol{W},\xi} \sum_{i=1}^{l} ||\boldsymbol{w}_i||^2 + \lambda_1 \sum_{i=1}^{m}\sum_{(u,v)} \xi_i^{uv} + \lambda_2 \sum_{j=1}^{l}\sum_{(a,b)} \xi_{ab}^{j} \\ \text{s.t. } \boldsymbol{w}_u^\top \boldsymbol{x}_i - \boldsymbol{w}_v^\top \boldsymbol{x}_i > 1 - \xi_i^{uv},\ \ \xi_i^{uv} \geq 0, \\ \text{for } i = 1, \cdots, m \text{ and } (u,v) \in Y_{i\cdot}^{+} \times Y_{i\cdot}^{-}\ , \\ \boldsymbol{w}_j^\top \boldsymbol{x}_a - \boldsymbol{w}_j^\top \boldsymbol{x}_b > 1 - \xi_{ab}^{j},\ \ \xi_{ab}^{j} \geq 0, \\ \text{for } j = 1, \cdots, l \text{ and } (a,b) \in Y_{\cdot j}^{+} \times Y_{\cdot j}^{-}\ .\end{aligned} \tag{3}$$

Here ξ_i^{uv} and ξ_{ab}^{j} are the slack variables, and λ_1, λ_2 are the trade-off parameters. When both λ_1 and λ_2 are positive, both label-wise and instance-wise margins are considered. If we set $\lambda_1 = 0$ (or $\lambda_2 = 0$), then only the instance-wise (or label-wise) margin is considered. In this paper, if the approach only considers instance-wise (or label-wise) margin , we call the approach as LIMO-inst (or LIMO-label). And LIMO considers both the two margins.

Algorithm 1 LIMO

Input:

Data matrix $\boldsymbol{X} \in \mathbb{R}^{m\times d}$, label matrix $\boldsymbol{Y} \in \{0,1\}^{m\times l}$, step size η, trade-off parameters λ_1, λ_2, and the maximium iteration number T.

Procedure:

Initialize $\boldsymbol{W}^0$ with $N(0, 1/\sqrt{d})$ random values.
Compute the weight vector $\boldsymbol{c}^{inst}$ of each instance, $\boldsymbol{c}_i^{inst} = |Y_{i\cdot}^{+}||Y_{i\cdot}^{-}| / \sum_{i=1}^{m} |Y_{i\cdot}^{+}||Y_{i\cdot}^{-}|$.
Compute the weight vector $\boldsymbol{c}^{label}$ of each label, $\boldsymbol{c}_j^{label} = |Y_{\cdot j}^{+}||Y_{\cdot j}^{-}| / \sum_{j=1}^{l} |Y_{\cdot j}^{+}||Y_{\cdot j}^{-}|$.
for $t = 1, 2, \cdots, T$ **do**
  Random sample an instance $\boldsymbol{x}_i^t$ using weight $\boldsymbol{c}^{inst}$,
  Random sample a positive label y_{iu} and a negative label y_{iv} of instance $\boldsymbol{x}_i^t$.
  if $1 - \boldsymbol{w}_u^\top \boldsymbol{x}_i^t + \boldsymbol{w}_v^\top \boldsymbol{x}_i^t > 0$ **then**
    $\boldsymbol{w}_u^t = \boldsymbol{w}_u^{t-1} - \eta(-\lambda_1 \boldsymbol{x}_i^t + \boldsymbol{w}_u^{t-1})$.
    $\boldsymbol{w}_v^t = \boldsymbol{w}_v^{t-1} - \eta(\lambda_1 \boldsymbol{x}_i^t + \boldsymbol{w}_v^{t-1})$.
  end if
  Random sample index j of label using weight $\boldsymbol{c}^{label}$.
  Random sample a positive instance $\boldsymbol{x}_a^t$ and a negative instance $\boldsymbol{x}_b^t$ on label j.
  if $1 - \boldsymbol{w}_j^\top \boldsymbol{x}_a^t + \boldsymbol{w}_j^\top \boldsymbol{x}_b^t > 0$ **then**
    $\boldsymbol{w}_j^t = \boldsymbol{w}_j^{t-1} - \eta(\lambda_2(\boldsymbol{x}_b^t - \boldsymbol{x}_a^t) + \boldsymbol{w}_j^{t-1})$.
  end if
end for
$\boldsymbol{W} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{W}^t$.

Output:

Multi-label linear model $\boldsymbol{W}$.

4.2. Algorithm

The objective Equation (3) is difficult to solve directly because of the large number of constraints and slack variables. For a training set with m instances and l labels, the number of constraints will be $O(m^2l + ml^2)$, which may exceed memory limit in real-world applicaitons.

In order to deal with the computational problem, we solve Equation (3) by stochastic gradient descent (SGD) with fixed step size and the default averaging technique in Shalev-Shwartz & Ben-David (2014, Chapter 14.3). The key point of SGD is to find out a random vector, whose expected value at each iteration equals the gradient direction. We randomly sample two kinds of triplets and use them to compute the correct direction. At each iteration t, we sample a triplet $(\boldsymbol{x}_i^t, y_{iu}, y_{iv})$ where y_{iu} is relevant and y_{iv} is irrelevant, and a triplet $(j, \boldsymbol{x}_a^t, \boldsymbol{x}_b^t)$ where $\boldsymbol{x}_a^t$ is a positive instance and $\boldsymbol{x}_b^t$ is a negative instance both on label j. Then we use the two triplets to compute the random gradient vector for SGD. The detailed algorithm is presented in Algorithm 1 and the proof that the random vector is an unbiased estimation of the gradient direction is available in Appendix A.3.

After the training procedure, we can use the linear model to predict continuous confidence values on the training data, then choose the best threshold value by optimizing a specific classification measure.

5. Experiments

We conduct experiments with LIMO on both synthetic and benchmark data. Note that the main purpose of our work is to study multi-label performance measures from the aspect of margin optimization, and thus, the goal of our experiments is to validate our theoretical findings rather than claim that LIMO is superior, although its performance is really highly competitive.

5.1. Synthetic Data

We conduct experiments on synthetic data with 4 labels. 2000 data points are randomly generated from a $(-1, +1)^2$ square, and the labels are assigned as in Figure 1. 50% data are held out for testing. The synthetic data is designed to simulate a typical real-world circumstance. The number of co-occurrent labels varies, the regions of each label are different and the data cannot be perfectly seperated by a linear learner.

To demonstrate the relationship between margins and performance measures, we degenerate LIMO to only consider either margin by setting the trade-off parameter λ_1 or λ_2 to zero. LIMO-inst sets $\lambda_1 = 0$ and LIMO-label sets $\lambda_2 = 0$. The other parameter is set to 100 and LIMO sets

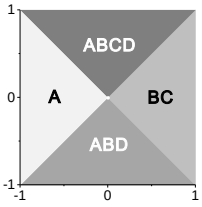

Figure 1. Input space consists of four regions with different assignments of the label set $\{A, B, C, D\}$. The center point is with coordinate $(0, 0)$.

$\lambda_1 = \lambda_2 = 100$. Ten replications of the experiment are conducted and the average results are reported. Because the range of performance measure coverage is not $[0, 1]$, while some performance measures are better when higher, and some are better when lower, we rescale all the performance values into relative values for clearer visualization. The best one is rescaled to 1 and the worst one is rescaled to 0. Figure 2 shows the relative results, where the originally worst performance value is given on the right.

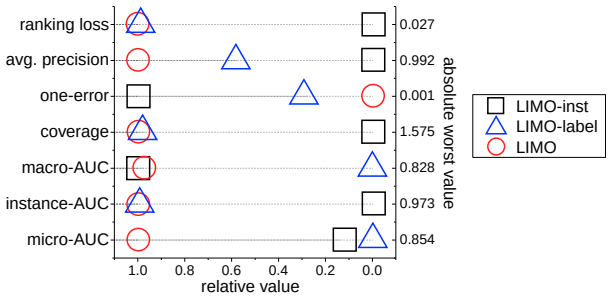

Figure 2. Summary of the relative performance on ranking measures. The more to the left, the better the performance.

The results shown in Figure 2 support our theoretical findings in Table 2. For example, micro-AUC is considered to be optimized by double effective F but not the other two, therefore LIMO (the red circle) gets the best relative value. For some measures proved to be optimized by label-wise margin such as ranking loss, average precision, coverage and instance-AUC, LIMO-label beats LIMO-inst. While for macro-AUC, LIMO-inst wins. For one-error, all three versions of LIMO do extremely well and get less than 0.001 absolute value, which may be the reason why the relative values are unexpected.

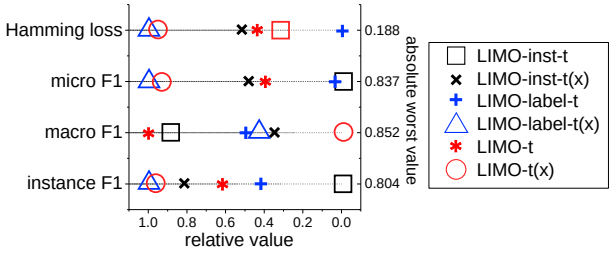

Figure 3. Summary of the relative performance on classification measures. The more to the left, the better the performance.

Figure 3 shows the relative performance on classification. We use two types of thresholding discussed in Section 3.2:

threshold function based on each instance or each label (denoted by -t(x) or -t in the legend). The thresholds are estimated on training data. This figure exactly shows our theoretical results: LIMO-label equipped with $t(x)$ can optimize instance-F1 and micro-F1; LIMO-inst equipped with t can optimize macro-F1. By considering both label-wise margin and instance-wise margin, LIMO works well on all four classificaiton measures.

5.2. Benchmark Data

We conduct experiments on eleven multi-label performance measures to further show that optimizing the label-wise or the instance-wise margin can lead to different results, as revealed in our theoretical analysis.

Five benchmark multi-label datasets[1] are used in our experiments. We choose them because they denote different domains: (i) A music dataset **CAL500**, (ii) an email dataset **enron**, (iii) a clinical text dataset **medical**, (iv) an image dataset **corel5k**, (v) a tagging dataset **bibtex**. We randomly split each dataset into two parts, i.e., 70% for training and 30% for testing. The experiments are repeated ten times, and the averaged results are reported.

Because our algorithm optimizes a linear model, three linear methods called Binary Relevance (BR) (Zhang & Zhou, 2014), ML-kNN (Zhang & Zhou, 2007) and GFM (Waegeman et al., 2014) are provided for fair comparison. As in experiments on synthetic data, we degenerate LIMO ($\lambda_1 = \lambda_2 = 1$) to LIMO-inst ($\lambda_1 = 0, \lambda_2 = 1$) and LIMO-label ($\lambda_1 = 1, \lambda_2 = 0$). The step size of SGD is set to 0.01. For BR, L2-regularized SVM (Chang & Lin, 2011) with C=1 is used as base learner. For ML-kNN and GFM, the number of nearest neighbors is 10. Suitable thresholds discussed in Section 3.2 are used for classification measures. We take the default parameter settings recommended by authors of the compared methods respectively. Because on one hand, we believe the parameter settings recommended by their authors are meaningful, on the other hand, it is hard to say which parameter setting is better in terms of eleven performance measures.

Because some measures are better when higher, and some measures are better when lower, to demonstrate the results more clearly, we compute the average rank of each approach over all datasets on a specific measure. For example, when we want to examine how LIMO performs on ranking loss, we first compute the ranks on each dataset: LIMO ranks 1st on CAL500, enron, bibtex and ranks 2nd on medical, corel5k. Then the average rank of LIMO on ranking loss is (1+1+1+2+2)/5=1.4. Figure 4 shows the average ranks. Due to the space limit, the detailed results used to compute the ranks are provided in Appendix B.2.

[1] http://mulan.sourceforge.net/datasets-mlc.html

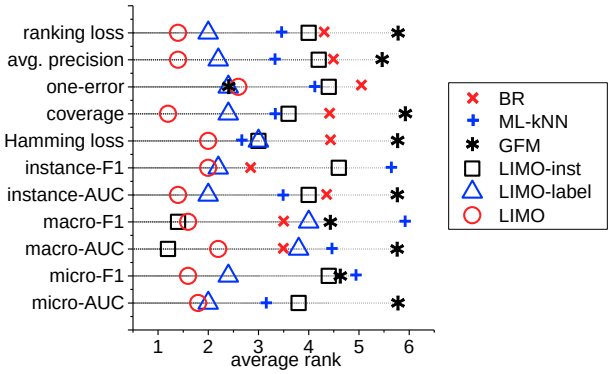

Figure 4. Average rank on benchmark data. The smaller the rank value, the better the performance.

The results in Figure 4 are consistent with our theoretical findings. LIMO-inst (the square) performs well on marco-F1 and macro-AUC, while LIMO-label (the triangle) performs well on other performance measures. LIMO (the circle) almost ranks top on every performance measure.

The experiments on synthetic and benchmark data support our theoretical analysis. Although different performance measures focus on different aspects, they share the common property which is formalized in our work as label-wise margin and instance-wise margin. In practice, it is recommended to use higher weight (λ_1/λ_2) on specific margin to optimize the required performance measure. LIMO with nonlinear predictors may perform better, which needs a novel optimization algorithm.

6. Conclusion

In this paper, we establish a unified view for a variety of multi-label performance measures. Based on the proposed concepts of label-wise/instance-wise margins, we prove that some performance measures are to be optimized by label-wise effective classifiers, whereas some by instance-wise effective classifiers. Inspired by the theoretical findings, we design the LIMO approach which can be adjusted to label-wise/instance-wise effective via different parameter settings.

Our work discloses that there are some shared properties among different subsets of multi-label performance measures. This explains why some measures seem to be redundant in experiments, and suggests that in future empirical studies, rather than randomly grasp a set of measures for evaluation, it is more informative to evaluate using measures with different properties, such as some measures optimized by label-wise effective predictors and some optimized by instance-wise effective predictors. In the future, it is encouraging to study the asymptotic properties of these performance measures when the two margins are suboptimal. The margin view also sheds a light for the design of novel multi-label algorithms.

Acknowledgements

This research was supported by the NSFC (61333014), 973 Program (2014CB340501), and the Collaborative Innovation Center of Novel Software Technology and Industrialization. Authors want to thank reviewers for helpful comments, and thank Sheng-Jun Huang, Xiu-Shen Wei, Miao Xu for reading a draft.

References

Boutell, Matthew R., Luo, Jiebo, Shen, Xipeng, and Brown, Christopher M. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

Chang, Chih-Chung and Lin, Chih-Jen. LIBSVM: A library for support vector machines. *ACM Trans. Intelligent Systems and Technology*, 2(3):27:1–27:27, 2011.

Dembczynski, Krzysztof, Waegeman, Willem, Cheng, Weiwei, and Hüllermeier, Eyke. Regret analysis for performance metrics in multi-label classification: the case of hamming and subset zero-one loss. In *ECML/PKDD*, pp. 280–295. 2010.

Elisseeff, André and Weston, Jason. A kernel method for multi-labelled classification. In *NIPS*, pp. 681–687, 2002.

Fan, Rong-En and Lin, Chih-Jen. A study on threshold selection for multi-label classification. *National Taiwan University, Technical Report*, pp. 1–23, 2007.

Fürnkranz, Johannes, Hüllermeier, Eyke, Mencía, Eneldo Loza, and Brinker, Klaus. Multilabel classification via calibrated label ranking. *Machine Learning*, 73 (2):133–153, 2008.

Gao, Wei and Zhou, Zhi-Hua. On the consistency of multi-label learning. *Artificial Intelligence*, 199:22–44, 2013.

Huang, Sheng-Jun, Yu, Yang, and Zhou, Zhi-Hua. Multi-label hypothesis reuse. In *ACM SIGKDD*, pp. 525–533, 2012.

Koyejo, Oluwasanmi, Natarajan, Nagarajan, Ravikumar, Pradeep, and Dhillon, Inderjit S. Consistent multilabel classification. In *NIPS*, pp. 3321–3329, 2015.

Lampert, Christoph H. Maximum margin multi-label structured prediction. In *NIPS*, pp. 289–297, 2011.

Liu, Weiwei and Tsang, Ivor W. On the optimality of classifier chain for multi-label classification. In *NIPS*, pp. 712–720, 2015.

Pham, Anh T., Raich, Raviv, Fern, Xiaoli Z., and Arriaga, Jesús Pérez. Multi-instance multi-label learning in the presence of novel class instances. In *ICML*, pp. 2427–2435, 2015.

Read, Jesse, Pfahringer, Bernhard, Holmes, Geoff, and Frank, Eibe. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011.

Schapire, Robert E and Singer, Yoram. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2):135–168, 2000.

Shalev-Shwartz, Shai and Ben-David, Shai. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Tsoumakas, Grigorios, Katakis, Ioannis, and Vlahavas, Ioannis P. Random k-labelsets for multilabel classification. *IEEE Trans. Knowledge and Data Engineering*, 23 (7):1079–1089, 2011.

Turnbull, Douglas, Barrington, Luke, Torres, David A., and Lanckriet, Gert R. G. Semantic annotation and retrieval of music and sound effects. *IEEE Trans. Audio, Speech & Language Processing*, 16(2):467–476, 2008.

Waegeman, Willem, Dembczynski, Krzysztof, Jachnik, Arkadiusz, Cheng, Weiwei, and Hüllermeier, Eyke. On the bayes-optimality of F-measure maximizers. *Journal of Machine Learning Research*, 15(1):3333–3388, 2014.

Ye, Nan, Chai, Kian Ming Adam, Lee, Wee Sun, and Chieu, Hai Leong. Optimizing F-measure: A tale of two approaches. In *ICML*, 2012.

Zhang, Min-Ling and Wu, Lei. LIFT: Multi-label learning with label-specific features. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 37(1):107–120, 2015.

Zhang, Min-Ling and Zhou, Zhi-Hua. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

Zhang, Min-Ling and Zhou, Zhi-Hua. A review on multi-label learning algorithms. *IEEE Trans. Knowledge and Data Engineering*, 26(8):1819–1837, 2014.