Optimizing Evaluation Metrics for Multi-Task Learning via the Alternating Direction Method of Multipliers

Ge-Yang Ke, Yan Pan, Jian Yin, Chang-Qin Huang

Abstract—Multi-task learning (MTL) aims to improve the generalization performance of multiple tasks by exploiting the shared factors among them. Various metrics (e.g., F-score, Area Under the ROC Curve) are used to evaluate the performances of MTL methods. Most existing MTL methods try to minimize either the misclassified errors for classification or the mean squared errors for regression. In this paper, we propose a method to directly optimize the evaluation metrics for a large family of MTL problems. The formulation of MTL that directly optimizes evaluation metrics is the combination of two parts: (1) a regularizer defined on the weight matrix over all tasks, in order to capture the relatedness of these tasks; (2) a sum of multiple structured hinge losses, each corresponding to a surrogate of some evaluation metric on one task. This formulation is challenging in optimization because both of its parts are non-smooth. To tackle this issue, we propose a novel optimization procedure based on the alternating direction scheme of multipliers, where we decompose the whole optimization problem into a sub-problem corresponding to the regularizer and another sub-problem corresponding to the structured hinge losses. For a large family of MTL problems, the first sub-problem has closed-form solutions. To solve the second sub-problem, we propose an efficient primal-dual algorithm via coordinate ascent. Extensive evaluation results demonstrate that, in a large family of MTL problems, the proposed MTL method of directly optimization evaluation metrics has superior performance gains against the corresponding baseline methods.

Index Terms—Multi-Task Learning, Evaluation Metrics, Structured Outputs, Coordinate Ascent, Alternating Direction Method of Multipliers.

✦

1 INTRODUCTION

Recently, considerable research has been devoted to *Multi-Task Learning (MTL)*, a problem of improving the generalization performance of multiple tasks by utilizing the shared information among them. MTL has been widely-used in various applications, such as natural language processing [1], handwritten character recognition [30], [34], scene recognition [29] and medical diagnosis [3]. Many MTL methods have been proposed in the literature [8], [11], [49], [51], [13], [21], [28], [30], [53], [1], [9], [10], [33], [29], [15], [46], [18], [52], [26].

In this paper, we consider MTL for classification or regression problems. Note that either a multi-class classification problem or a multi-label learning problem can be regarded as an MTL problem[1]. Most of the existing MTL methods focus on minimizing either a convex surrogate (e.g. the hinge loss or the logistic loss) of the 0-1 errors for multi-task classification, or the mean squared errors for multi-task regression. On the other hand, in practice, several evaluation metrics other than misclassified errors or mean squared errors are used the evaluation of MTL methods, e.g., F-score, Precision, Recall, Area Under the ROC Curve (AUC), Mean Average Precision. For example, in the cases of MTL on imbalanced data (e.g., in a task, the number of negative samples is much larger than that of the positive samples), cost-sensitive MTL or MTL for ranking, these metrics are more effective in performance evaluation than the standard misclassified errors or the mean squared errors. However, due to the computational difficulties, few learning techniques have been developed to directly optimize these evaluation metrics in MTL.

In this paper, we propose an approach to directly optimizing the evaluation metrics in MTL, which can be applied to a large family of MTL problems. Specifically, for an MTL problem with m tasks (the ith task is associated with a training set $\{(\mathbf{x}_j^{(i)}, \mathbf{y}_j^{(i)})\}_{j=1}^{n_i}$, $i = 1, 2, ..., m$, n_i represents the number of training samples for the ith

---

Ge-Yang Ke, Yan Pan, and Jian Yin are with the School of Data Science and Computer Science, Sun Yat-sen University, Guangzhou 510006, China. Corresponding author: Yan Pan (panyan5@mail.sysu.edu.cn)
Chang-Qin Huang is with the School of Information Technology in Education, South China Normal University, Guangzhou, 510631, China

1. As an illustrative example, we consider a multi-label classification problem with instances $\{x_1, x_2, x_3, x_4, x_5\}$ that x_1 belongs to classes a and b, x_2 belongs to classes b and c, x_3 belongs to class c, x_4 belongs to class a, x_5 belongs to classes a, b and c. This problem can be regarded as an MTL problem with three tasks, where the training sets for each of these tasks are:

$$(x_1, 1), (x_2, 0), (x_3, 0), (x_4, 1), (x_5, 1)$$
$$(x_1, 1), (x_2, 1), (x_3, 0), (x_4, 0), (x_5, 1)$$
$$(x_1, 0), (x_2, 1), (x_3, 1), (x_4, 0), (x_5, 1)$$

The first/second/third task is a binary classification problem of an instance belonging to class a/ b/c or not. Hence, a multi-label learning problem is a special case of an MTL problem. Similarly, we can verify that a multi-class classification problem can also be regarded as an MTL problem.

task), we consider a generic formulation in the following:

$$\min_{\mathbf{W}} \Omega(\mathbf{W}) + \lambda \sum_{i=1}^{m} \mathcal{L}(\mathbf{W}_{.i}; \{(\mathbf{x}_j^{(i)}, \mathbf{y}_j^{(i)})\}_{j=1}^{n_i}), \quad (1)$$

where $\mathbf{W}$ is the weight matrix with m columns $\mathbf{W}_{.1}$, $\mathbf{W}_{.2}$, ..., $\mathbf{W}_{.m}$, $\lambda > 0$ is a trade-off parameter. This formulation is the linear combination of two parts. The first part is a regularizer $\Omega(\mathbf{W})$ defined on the weight matrix $\mathbf{W}$ over all tasks, in order to leverage the relatedness of these tasks. Examples of this kind of regularizer include the trace-norm, the $\ell_{1,1}$-norm or the $\ell_{2,1}$-norm on $\mathbf{W}$. The second part in the formulation is a sum of multiple loss functions, each corresponds to one task. In order to directly optimize a specific evaluation metric, we consider the hinge loss functions for structured outputs [39], [20], [50], [48], [47], which are surrogates of a specific evaluation metric.

Such a formulation in (1) includes a large family of MTL problems. Since the two parts in (1) are usually non-smooth, the optimization problem (1) is difficult to solve. To tackle this issue, we propose a novel optimization procedure based on the alternating direction scheme of multipliers (ADMM [6], [25]), which is widely used in various machine learning problems (e.g., [31], [32], [33], [44]). We decompose the whole optimization problem in (1) into two simpler sub-problems. The first sub-problem corresponds to the regularizer. For commonly-used regularizers (e.g., the trace-norm, the $\ell_{2,1}$-norm) in MTL, this sub-problem can be solved by close-form solutions. The second sub-problem corresponds to the structured hinge losses. To solve the second sub-problem, we propose an efficient primal-dual algorithm via coordinate ascent.

We conduct extensive experiments to evaluate the performances of the proposed MTL method. Experimental results show that the proposed method that optimizes a specific evaluation metric outperforms the corresponding MTL classification or MTL regression baseline methods by a clear margin.

2 RELATED WORK

MTL is a wide class of learning problems. Roughly speaking, the existing MTL methods can be divided into three main categories: parameters sharing, common features sharing, and low-rank subspace sharing.

In the methods with parameter sharing, all tasks are assumed to explicitly share some common parameters. Representative methods in this category include shared weight vectors [11], hidden units in neural network [8], and common prior in Bayessian models [49], [51].

In the methods with common features sharing, task relatedness is modeled by enforcing all tasks to share a common set of features [2], [28], [22], [30], [13], [21], [53]. Representative examples are the methods which constrain the model parameters (i.e., a weight matrix) of all tasks to have certain sparsity patterns, for example, cardinality sparsity [30], group sparsity [28], [13], or clustered structure [21], [53].

The methods in the third category assume that all tasks lie in a shared low-rank subspace [1], [9], [10]. A common assumption in these category of methods is that most of the tasks are relevant while (optionally) there may exist a small number of irrelevant (outlier) tasks. These methods pursue a low-rank weight matrix that captures the underlying shared factors among tasks. Trace-norm regularization is commonly-used in these methods to encourage the low-rank structure on the model parameters.

Most of the existing MTL methods are focused on designing regularizers or parameter sharing patterns to utilize the intrinsic relationships among multiple related tasks. These MTL methods usually try to optimize the classification errors or the mean squared errors for regression. In practice, various other metrics (such as F-score and AUC) are used in the evaluation of MTL methods. However, little effort has been devoted to optimize these evaluation metrics in the context of MTL except for the work [14], in which the author proposed a hierarchical MTL formulation for structured output prediction in sequence segmentation. Since the regularizer used in [14] is decomposable, the hierarchical MTL problem can be decomposed into multiple independent tasks, each is a structure output learning problem with a simple regularizer. In this paper, we seek to directly optimize commonly-used evaluation metrics for MTL with possibly indecomposable regularizer, resulting in a generic approach that can be applied to a large family of MTL problems. Our formulation can be regarded as MTL for structure output prediction with an indecomposable regularizer.

The proposed methods in this paper are also related to the multi-label algorithms. There are various multi-label algorithm proposed in the literature, e.g., the RAkEL method that uses random k-label sets [41], the MLCSSP method that spans the original label space by subset of labels [4], the AdaBoostMH method based on AdaBoost [37], the HOMER method based on the hierarchy of multi-label learners [40], the binary relevance (BR) [42] method, the label power-set (LP) [42] method, and the ensembles of classifier chains (ECC) [35] method.

The proposed approach in this paper is to optimize the evaluation metrics in MTL. We refer the readers to Section 4 for the detailed introduction to the evaluation metrics related to the proposed approach.

3 NOTATIONS

We first introduce the notations to be used throughout this paper. We use bold upper-case characters (e.g., $\mathbf{M}$, $\mathbf{X}$, $\mathbf{W}$) to represent matrices, and bold lower-case characters (e.g., $\mathbf{x}$, $\mathbf{y}$) to represent vectors, respectively. For a matrix $\mathbf{M} \in \mathbb{R}^{d \times m}$, we denote $\mathbf{M}_{ij}$ as the the element at the cross of the ith row and jth column in $\mathbf{M}$. We denote $\mathbf{M}_{i.} \in \mathbb{R}^{1 \times m}$ as the ith row of $\mathbf{M}$, and $\mathbf{M}_{.j} \in \mathbb{R}^{d \times 1}$ as the j-th column of $\mathbf{M}$, respectively.

We denote $||\mathbf{M}||_F$ as the Frobenius norm of $\mathbf{M}$ that $||\mathbf{M}||_F = \sqrt{\sum_{i=1}^d \sum_{j=1}^m (\mathbf{M}_{ij})^2}$. Let $||\mathbf{M}||_{1,1} = \sum_{i=1}^d \sum_{j=1}^m |\mathbf{M}_{ij}|$ be the $\ell_{1,1}$-norm of $\mathbf{M}$, where $|\mathbf{M}_{ij}|$ is the absolute value of $\mathbf{M}_{ij}$. Let $||\mathbf{M}||_{2,1} = \sum_{i=1}^d ||\mathbf{M}_{i.}||_2$ be the $\ell_{2,1}$-norm of $\mathbf{M}$, where $||\mathbf{M}_{i.}||_2 = \sqrt{\sum_{j=1}^m \mathbf{M}_{ij}^2}$ is the ℓ_2-norm of $\mathbf{M}_{i.}$. Let $||\mathbf{M}||_\infty = \max_{i,j} |\mathbf{M}_{ij}|$ be the infinity norm of $\mathbf{M}$. The trace-norm of $\mathbf{M}$ is defined by $||\mathbf{M}||_* = \sum_{k=1}^{rank(\mathbf{M})} \sigma_k(\mathbf{M})$, where $\{\sigma_k(\mathbf{M})\}_{k=1}^{rank(\mathbf{M})}$ are the non-zero singular values of $\mathbf{M}$ and $rank(\mathbf{M})$ is the rank of $\mathbf{M}$. We denote $\mathbf{M}^T$ as the transpose of $\mathbf{M}$. For a vector $\mathbf{x}$, $||\mathbf{x}||_2$ represent the ℓ_2-norm.

In the context of MTL, we assume we are given m learning tasks. The ith ($i = 1, \ldots, m$) task is associated with a training set $(\mathbf{X}^{(i)}, \mathbf{y}^{(i)})$, where $\mathbf{X}^{(i)} \in \mathbb{R}^{n_i \times d}$ denotes the data matrix with each row being a sample, $\mathbf{y}^{(i)} \in \{-1, +1\}^{n_i}$ denotes the target labels on $\mathbf{X}^{(i)}$, d is the feature dimensionality, and n_i is the number of samples for the ith task. For $i = 1, 2, ..., m$, we define $\mathbb{E}_i = \{-1, +1\}^{n_i}$ as the set of all possible n_i-dimension vector, each of whose elements is either -1 or 1. To simplify presentation, we assume $\mathbb{E}_i = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_p\}$ where $p = 2^{n_i}$ and $\mathbf{y}_j$ is one of the possible vectors that belong to $\{-1, 1\}^{n_i}$.

We define a weight matrix $\mathbf{W} = [\mathbf{W}_{.1}, \ldots, \mathbf{W}_{.m}] \in \mathbb{R}^{d \times m}$ on all of the m tasks. The goal of (linear) MTL is to simultaneously learn m (linear) predictors $\mathbf{W}_{.i}$ ($i = 1, \ldots, m$) to minimize some loss function $\mathcal{L}(\mathbf{W}_{.i}; \mathbf{X}^{(i)}, \mathbf{y}^{(i)})$ (e.g. the least square loss $||\mathbf{y}^{(i)} - \mathbf{X}^{(i)} \mathbf{W}_{.i}||_2^2$), where $\mathbf{W}_{.i} \in \mathbb{R}^d$ is in the form of a column vector. Note that for each task, we have $\mathbf{X}^{(i)} = [\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \cdots, \mathbf{x}_{n_i}^{(i)}]^T$ and $\mathbf{y}^{(i)} = [\mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)}, \cdots, \mathbf{y}_{n_i}^{(i)}]^T$.

4 PROBLEM FORMULATIONS

The linear MTL problem can be formulated as the generic form in (1). The objective functions in many existing MTL methods are special cases of such a formulation. The following are two examples:

- With the regularizer $\Omega(\mathbf{W})$ being the $\ell_{2,1}$-norm $||\mathbf{W}||_{2,1}$ and each loss function $\mathcal{L}(\mathbf{W}_{.i}; \mathbf{X}^{(i)}, \mathbf{y}^{(i)})$ being the mean squared loss $\frac{1}{2}||\mathbf{y}^{(i)} - \mathbf{X}^{(i)} \mathbf{W}_{.i}||_2^2$, the problem in (1) is the same as the objective used in [28].
- If the regularizer $\Omega(\mathbf{W})$ is set to be the trace-norm $||\mathbf{W}||_*$ and each loss function $\mathcal{L}(\mathbf{W}_{.i}; \mathbf{X}^{(i)}, \mathbf{y}^{(i)})$ is smooth (e.g., the mean squared loss $\frac{1}{2}||\mathbf{y}^{(i)} - \mathbf{X}^{(i)} \mathbf{W}_{.i}||_2^2$), the problem in (1) becomes the objective used in [17].

The existing MTL methods mainly focus on the design of good regularizers (i.e., $\Omega(\mathbf{W})$) to catch the shared factors among multiple related tasks. The loss functions used in these methods are either to minimize the misclassified errors (for classification) or the mean squared errors (for regression). On the other hand, in practice, several evaluation metrics other than misclassified errors or mean squared errors are used the evaluation of MTL

methods, such as F-score and AUC. Particularly, in the cases of MTL on imbalanced data (e.g., in a task, the number of negative samples is much larger than that of the positive samples), these metrics are more effective in performance evaluation than the standard misclassified errors or the mean squared errors.

Learning techniques of directly optimizing evaluation metrics, as known as learning with structured outputs, have been developed for many (single-task) problems, e.g., classification [39], [20], ranking [50]. However, despite the acknowledged importance of the metrics like F-score or AUC, little effort has been made to design MTL methods that directly optimize these evaluation metrics. The main reason is that MTL of optimizing the evaluation metrics usually results in a non-smooth objective function which is difficult to solve.

In this paper, we focus on MTL with structured outputs and propose a generic optimization procedure based on ADMM. This optimization procedure can be applied to solving a large family of MTL problems that directly optimize some evaluation metric (e.g., F-score, AUC). We call the proposed method Structured MTL (SMTL for short).

The formulation of SMTL is also a special case of (1). In order to optimize some evaluation metric, we define the loss function for each task as the structured hinge loss:

$$\mathcal{L}(\mathbf{W}_{.i}; \mathbf{X}^{(i)}, \mathbf{y}^{(i)})$$
$$= \max_{\mathbf{y}_j \in \mathbb{E}_i} [\Delta(\mathbf{y}^{(i)}, \mathbf{y}_j) - \mathbf{W}_{.i}^T \mathbf{X}^{(i)^T} (\mathbf{y}^{(i)} - \mathbf{y}_j)],$$

where $\mathbf{y}_j$ represents any possible label assignment on $\mathbf{X}^{(i)}$. $\Delta(\mathbf{y}^{(i)}, \mathbf{y}_j)$ represents an evaluation metric to measure the distance between the true labels $\mathbf{y}^{(i)}$ and the other labels $\mathbf{y}_j$. For example, $\Delta(., .)$ can be 1-F-score or 1-AUC.

The formulation of SMTL is defined as:

$$\min_{\mathbf{W}} \Omega(\mathbf{W})$$
$$+ \lambda \sum_{i=1}^m \max_{\mathbf{y}_j \in \mathbb{E}_i} [\Delta(\mathbf{y}^{(i)}, \mathbf{y}_j) - \mathbf{W}_{.i}^T \mathbf{X}^{(i)^T} (\mathbf{y}^{(i)} - \mathbf{y}_j)]. \quad (2)$$

In this paper, we only focus on the MTL problems in the form of (2) that satisfy the following conditions:

- **Condition 1:** With respect to $\Omega(\mathbf{W})$, there is a close-form solution for the following sub-problem

$$\min_{\mathbf{W}} \Omega(\mathbf{W}) + \frac{\mu}{2} ||\mathbf{W} - \mathbf{M}||_F^2 \quad (3)$$

where $\mathbf{M} \in \mathbb{R}^{d \times m}$ and μ is a positive constant.
- **Condition 2:** For the evaluation metric $\Delta(\mathbf{y}^{(i)}, \mathbf{y}_j)$, the following sub-problem can be solve in polynomial time.

$$\operatorname*{argmax}_{\mathbf{y}_j \in \mathbb{E}_i} [\Delta(\mathbf{y}^{(i)}, \mathbf{y}_j) - \mathbf{W}_{.i}^T \mathbf{X}^{(i)^T} (\mathbf{y}^{(i)} - \mathbf{y}_j)] \quad (4)$$

The first condition is to restrict the regularizer $\Omega(\mathbf{W})$ and the second one is to restrict the evaluation metric function $\Delta(\mathbf{y}^{(\mathbf{i})}, \mathbf{y}_j)$. Even under these conditions, the

formulation in (2) includes a large family of MTL problems. On the one hand, for the regularizer $\Omega(\mathbf{W})$, the following norms that are commonly-used in MTL satisfy Condition 1:

- $\ell_{1,1}$**-norm** For the MTL problems with $\Omega(\mathbf{W}) = ||\mathbf{W}||_{1,1}$, the sub-problem in (3) is known to have the close-form solution

$$\mathbf{W} = \mathcal{S}_{\frac{1}{\mu}}(\mathbf{M}), \tag{5}$$

where $\mathcal{S}_{\delta}(\mathbf{M}) = \max(\mathbf{M} - \delta, 0) + \min(\mathbf{M} + \delta, 0)$ is the shrinkage operator [25].

- $\ell_{2,1}$**-norm** For the MTL problems with $\Omega(\mathbf{W}) = ||\mathbf{W}||_{2,1}$, the sub-problem in (3) is also known to have close-form solutions:

$$\mathbf{W}_{j\cdot} = \begin{cases} \frac{||\mathbf{M}_{j\cdot}||_2 - \frac{1}{\mu}}{||\mathbf{M}_{j\cdot}||_2}\mathbf{M}_{j\cdot} & \text{if } \frac{1}{\mu} < ||\mathbf{M}_{j\cdot}||_2, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

- **Trace-norm** For the MTL problems with $\Omega(\mathbf{W}) = ||\mathbf{W}||_*$, the sub-problem in (3) is also have the close-form solution by the Singular Value Threshold method [7]. Specifically, by assuming $\mathbf{U\Sigma V}$ be the SVD form of $\mathbf{M}$, the close-form solution is given by:

$$\mathbf{W} = \mathbf{U}\mathcal{S}_{1/\mu}(\mathbf{\Sigma})\mathbf{V}^T. \tag{7}$$

On the other hand, many commonly-used metric functions satisfy the second condition. The following are two examples which we will consider in this paper:

- **MTL by directly optimizing F-Score** F-Score is a typical performance metric for binary classification, particularly in learning tasks on imbalanced data. F-Score is a trade-off between Precision and Recall. Specifically, given $\mathbf{y}^{(i)}$ and $\mathbf{y}_j$, we define the precision as:

$$Precision = \frac{\sum_{k=1}^{n_i} I(\mathbf{y}_k^{(i)} = 1 \text{ and } (\mathbf{y}_j)_k = 1)}{\sum_{k=1}^{n_i} I(\mathbf{y}_k^{(i)} = 1)},$$

and the recall as:

$$Recall = \frac{\sum_{k=1}^{n_i} I(\mathbf{y}_k^{(i)} = 1 \text{ and } (\mathbf{y}_j)_k = 1)}{\sum_{k=1}^{n_i} I((\mathbf{y}_j)_k = 1)},$$

where $I(condition)$ represents the indicator function that $I(condition) = 1$ if $condition$ is true, otherwise $I(condition) = 0$. Then the F-score on $\mathbf{y}^{(i)}$ and $\mathbf{y}_j$ is defined as:

$$F_\beta = \frac{(1 + \beta) \times Precision \times Recall}{Precision + \beta Recall}, \tag{8}$$

where β is a trade-off parameter. Hereafter, we simply set $\beta = 1$. Finally, the metric function $\Delta(.,.)$ with respect to the F-score is defined by:

$$\Delta(\mathbf{y}^{(i)}, \mathbf{y}_j) = 1 - F_\beta. \tag{9}$$

With such a form of $\Delta(\mathbf{y}^{(i)}, \mathbf{y}_j)$, the sub-problem in (4) can be solved in polynomial time [20].

- **MTL by directly optimizing AUC** AUC is also a popular performance metric for binary classification, particularly in imbalanced learning. Given $\mathbf{y}^{(i)}$ and $\mathbf{y}_j$, the AUC metric can be calculated by :

$$AUC = 1 - \frac{Swapped}{Pos \times Neg} \tag{10}$$

where $Swapped$ represents the number of "inverted" pairs in $\mathbf{y}^{(i)}$ compared to $\mathbf{y}_j$:

$$Swapped = \sum_{l=1}^{n_i} \sum_{k=1}^{n_i} I(\mathbf{y}_l^{(i)} = 1 \text{ and } \mathbf{y}_k^{(i)} = -1)$$
$$\times I((\mathbf{y}_j)_l = -1 \text{ and } (\mathbf{y}_j)_k = 1).$$

Pos/Neg represents the number of positive/negative samples in the ith task:

$$Pos = \sum_{k=1}^{n_i} I(\mathbf{y}_k^{(i)} = 1),$$

$$Neg = \sum_{k=1}^{n_i} I(\mathbf{y}_k^{(i)} = -1).$$

The corresponding $\Delta(.,.)$ can be defined as:

$$\Delta(\mathbf{y}^{(i)}, \mathbf{y}_j) = 1 - AUC. \tag{11}$$

With such a form of $\Delta(\mathbf{y}^{(i)}, \mathbf{y}_j)$, there also exists polynomial-time algorithms to solve the sub-problem in (4) [20].

Note that here the Precision, Recall, F-Score and AUC are defined for a particular task.

5 PROPOSED OPTIMIZATION PROCEDURE

5.1 Overview

In this section, we present the proposed optimization procedure to solve the problem (2). Our procedure is based on the scheme of ADMM.

For ease of presentation, we define

$$\mathcal{G}_i(\mathbf{W}_{\cdot i}) = \max_{\mathbf{y}_j}[\Delta(\mathbf{y}^{(i)}, \mathbf{y}_j) - \mathbf{W}_{\cdot i}^T \mathbf{X}^{(i)^T}(\mathbf{y}^{(i)} - \mathbf{y}_j)],$$

and

$$\mathcal{G}(\mathbf{W}) = \sum_{i=1}^m \mathcal{G}_i(\mathbf{W}_{\cdot i}).$$

Then, the problem in (2) can be re-formulated to its equivalent form in the following:

$$\begin{aligned} \min_{\mathbf{S}, \mathbf{W}} \quad & \Omega(\mathbf{S}) + \lambda \mathcal{G}(\mathbf{W}) \\ s.t \quad & \mathbf{W} - \mathbf{S} = 0, \end{aligned} \tag{12}$$

where $\mathbf{S} \in \mathbb{R}^{d \times m}$ is an auxiliary variable.

The corresponding augmented Lagrangian function with respect to (12) is:

$$\begin{aligned} \mathcal{A}(\mathbf{W}, \mathbf{S}, \mathbf{Z}) \\ = \Omega(\mathbf{S}) + \lambda \mathcal{G}(\mathbf{W}) + \langle \mathbf{Z}, \mathbf{W} - \mathbf{S} \rangle + \frac{\mu}{2}||\mathbf{W} - \mathbf{S}||_F^2 \end{aligned} \tag{13}$$

where $\mathbf{Z}$ is the Lagrangian multiplier, $\langle \cdot, \cdot \rangle$ represents the inner product of two matrices (i.e., given matrices $\mathbf{A}$ and $\mathbf{B}$, we have $\langle \mathbf{A}, \mathbf{B} \rangle = Tr(\mathbf{A}^T \mathbf{B})$, where $Tr(\mathbf{M})$ is the trace of the matrix $\mathbf{M}$), $\mu > 0$ is an adaptive penalty parameter.

Based on the ADMM scheme, the sketch of the proposed optimization procedure is shown in Algorithm 1, where in each iteration we alternatively update $\mathbf{W}$, $\mathbf{S}$ and $\mathbf{Z}$ by minimizing the Lagrangian function in (13) with other variables fixed. The update rules for $\mathbf{W}$, $\mathbf{S}$ and $\mathbf{Z}$ are in the following:

$$\mathbf{S}^{\{t+1\}} \leftarrow \underset{\mathbf{S}}{\operatorname{argmin}} \mathcal{A}(\mathbf{W}^{\{t\}}, \mathbf{S}, \mathbf{Z}^{\{t\}});$$
$$\mathbf{W}^{\{t+1\}} \leftarrow \underset{\mathbf{W}}{\operatorname{argmin}} \mathcal{A}(\mathbf{W}, \mathbf{S}^{\{t+1\}}, \mathbf{Z}^{\{t\}});$$
$$\mathbf{Z}^{\{t+1\}} \leftarrow \mathbf{Z}^{\{t\}} + \mu(\mathbf{W}^{\{t+1\}} - \mathbf{S}^{\{t+1\}}).$$

Note that hereafter we use $\mathbf{M}^{\{t\}}$ to represent the the value of variable $\mathbf{M}$ in the t-th iteration.

Next, we will present the details of solving the sub-problems with respect to $\mathbf{S}$ or $\mathbf{W}$, respectively, with other variables being fixed.

Algorithm 1 The proposed ADMM procedure for the structured MTL problem (2)

Input: training set $\{(\mathbf{X}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^m$, desired tolerant error ϵ, maximal iteration number T.
Output: Weight matrix $\mathbf{W} = [\mathbf{W}_{.1}, \cdots, \mathbf{W}_{.m}]$
1. Initialize: $\mathbf{Z} = \mathbf{S} = \mathbf{W} \leftarrow \mathbf{0}^{d \times m}$, $t \leftarrow 0$.
2. Repeat:
3.   Update
       $\mathbf{S}^{\{t+1\}} \leftarrow \underset{\mathbf{S}}{\operatorname{argmin}} \Omega(\mathbf{S}) + \frac{\mu}{2}||\mathbf{S} - \mathbf{W}^{\{t\}} - \mathbf{Z}^{\{t\}}/\mu||_F^2$
       by solving (15), (17) or (18) accordingly.
4.   For $i = 1$ to m
5.   Update $\mathbf{W}_{.i}^{\{t+1\}} \leftarrow$
         $\underset{\mathbf{W}_{.i}}{\operatorname{argmin}} \lambda \mathcal{G}_i(\mathbf{W}_{.i}) + \frac{\mu}{2}||\mathbf{W}_{.i} - \mathbf{S}_{.i}^{\{t+1\}} + \frac{\mathbf{Z}_{.i}^{\{t\}}}{\mu}||_2^2$
       by Algorithm 2.
6.   End For
7.   Update $\mathbf{Z}^{\{t+1\}} \leftarrow \mathbf{Z}^{\{t\}} + \mu(\mathbf{W}^{\{t+1\}} - \mathbf{S}^{\{t+1\}})$.
8. Until $||S - W||_\infty \leq \epsilon$ or $t = T$.

5.2 Solving the Sub-Problem for $\mathbf{S}$

In the t-th iteration (in the outer loop) of Algorithm 1, the sub-problem for $\mathbf{S}$ with respect to (13) can be simplified as:

$$\mathbf{S}^{\{t+1\}} \leftarrow \underset{\mathbf{S}}{\operatorname{argmin}} \mathcal{A}(\mathbf{W}^{\{t\}}, \mathbf{S}, \mathbf{Z}^{\{t\}})$$
$$= \underset{\mathbf{S}}{\operatorname{argmin}} \Omega(\mathbf{S}) + \frac{\mu}{2}\left\| \mathbf{W}^{\{t\}} - \mathbf{S} + \mathbf{Z}^{\{t\}}/\mu \right\|_F^2 \quad (14)$$

For different regularizer $\Omega(\mathbf{S})$, the solution to (14) is different.

- **Case 1: the $\ell_{1,1}$-norm** With $\Omega(\mathbf{S})$ being $||\mathbf{S}||_{1,1}$, by applying (5) to (14), we have:

$$\underset{\mathbf{S}}{\operatorname{argmin}}||\mathbf{S}||_{1,1} + \frac{\mu}{2}\left\| \mathbf{W}^{\{t\}} - \mathbf{S} + \mathbf{Z}^{\{t\}}/\mu \right\|_F^2$$
$$= \max(0, \mathbf{W}^{\{t\}} + \mathbf{Z}^{\{t\}}/\mu - 1/\mu) \quad (15)$$
$$+ \min(0, \mathbf{W}^{\{t\}} + \mathbf{Z}^{\{t\}}/\mu + 1/\mu).$$

- **Case 2: the $\ell_{2,1}$-norm** When $\Omega(\mathbf{S}) = ||\mathbf{S}||_{2,1}$, (14) can be rewritten as:

$$\underset{\mathbf{S}}{\operatorname{argmin}}||\mathbf{S}||_{2,1} + \frac{\mu}{2}\left\| \mathbf{W}^{\{t\}} - \mathbf{S} + \mathbf{Z}^{\{t\}}/\mu \right\|_F^2. \quad (16)$$

By applying (6) to (16), we obtain the following close-form solution:

$$\mathbf{S}_{j.} = \begin{cases} \frac{||\mathbf{M}_{j.}||_2 - \frac{1}{\mu}}{||\mathbf{M}_{j.}||_2} \mathbf{M}_{j.} & \text{if } \frac{1}{\mu} < ||\mathbf{M}_{j.}||_2, \\ 0 & \text{otherwise,} \end{cases} \quad (17)$$

where $\mathbf{M} = \mathbf{W}^{\{t\}} + \mathbf{Z}^{\{t\}}/\mu$.

- **Case 3: the trace-norm** When $\Omega(\mathbf{S}) = ||\mathbf{S}||_*$, we can apply (7) to (14) and obtain the following close-form solution:

$$\underset{\mathbf{S}}{\operatorname{argmin}}||\mathbf{S}||_* + \frac{\mu}{2}\left\| \mathbf{W}^{\{t\}} - \mathbf{S} + \mathbf{Z}^{\{t\}}/\mu \right\|_F^2.$$
$$= \mathbf{U}(\max(0, \mathbf{\Sigma} - 1/\mu) + \min(0, \mathbf{\Sigma} + 1/\mu))\mathbf{V}^T, \quad (18)$$

where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the SVD form of $\mathbf{W}^{\{t\}} + \mathbf{Z}^{\{t\}}/\mu$.

5.3 Solving the Sub-Problem for $\mathbf{W}$

5.3.1 Formulation

In the t-th outer iteration in Algorithm 1, the sub-problem for $\mathbf{W}$ with respect to (13) can be reformulated as:

$$\mathbf{W}^{\{t+1\}} \leftarrow \underset{\mathbf{W}}{\operatorname{argmin}} \mathcal{A}(\mathbf{W}, \mathbf{S}^{\{t+1\}}, \mathbf{Z}^{\{t\}})$$
$$= \underset{\mathbf{W}}{\operatorname{argmin}} \lambda \mathcal{G}(\mathbf{W}) + \frac{\mu}{2}\left\| \mathbf{W} - \mathbf{S}^{\{t+1\}} + \mathbf{Z}^{\{t\}}/\mu \right\|_F^2$$
$$= \underset{\mathbf{W}}{\operatorname{argmin}} \sum_{i=1}^m \lambda \mathcal{G}_i(\mathbf{W}_{.i}) + \frac{\mu}{2}\left\| \mathbf{W}_{.i} - \mathbf{S}_{.i}^{\{t+1\}} + \mathbf{Z}_{.i}^{\{t\}}/\mu \right\|_F^2 \quad (19)$$

To simplify presentation, we denote $\mathbf{b}_i = \mathbf{S}_{.i}^{\{t+1\}} - \mathbf{Z}_{.i}^{\{t\}}/\mu$. Then, the problem in (19) can be separated into m independent sub-tasks:

$$\underset{\mathbf{W}_{.i}}{\min} \lambda \mathcal{G}_i(\mathbf{W}_{.i}) + \frac{\mu}{2}\left\| \mathbf{W}_{.i} - \mathbf{b}_i \right\|_F^2, i = 1, ..., m. \quad (20)$$

For $j = 1, 2, ..., p$, we define $\mathbf{K} = [\mathbf{K}_{.1}, \mathbf{K}_{.2}, ..., \mathbf{K}_{.p}]$ with $\mathbf{K}_{.j} = \mathbf{X}^{(i)T}(\mathbf{y}^{(i)} - \mathbf{y}_j) + \frac{\mu}{\lambda}\mathbf{b}_i$, $\mathbf{\Delta} = (\mathbf{\Delta}_1, \mathbf{\Delta}_2, ..., \mathbf{\Delta}_p)^T$ with $\mathbf{\Delta}_j = \Delta(\mathbf{y}^{(i)}, \mathbf{y}_j)$. Then, the problem in (20) can be simplified as:

$$\underset{\mathbf{W}_{.i}}{\min} \lambda \mathcal{G}_i(\mathbf{W}_{.i}) + \frac{\mu}{2}\left\| \mathbf{W}_{.i} - \mathbf{b}_i \right\|_F^2$$
$$= \underset{\mathbf{W}_{.i}}{\min} \frac{\mu}{2}(||\mathbf{W}_{.i}||_F^2 + ||\mathbf{b}_i||_F^2 - 2\mathbf{W}_{.i}^T \mathbf{b}_i) \quad (21)$$
$$+ \lambda \underset{\mathbf{y}_j \in \mathbb{E}_i}{\max} [\Delta(\mathbf{y}^{(i)}, \mathbf{y}_j) - \mathbf{W}_{.i}^T \mathbf{X}^{(i)T}(\mathbf{y}^{(i)} - \mathbf{y}_j)].$$

By re-scaling the objective (21) by μ and drop the terms independent of $\mathbf{W}_{.i}$ and $\mathbf{y}_j$, we have:

$$\underset{\mathbf{W}_{.i}}{\min} \frac{1}{2}||\mathbf{W}_{.i}||_2^2 + \frac{\lambda}{\mu}\underset{j}{\max}[\mathbf{\Delta}_j - (\mathbf{W}_{.i}^T \mathbf{K})_j] \quad (22)$$

The existence of the max operator on exponential number of elements makes it difficult to optimize the objective in (22). To tackle this issue, in the next two subsection, we derive the Fenchel dual [36] form of (22) and develop a coordinate ascent algorithm to solve this dual form.

5.3.2 Fenchel Dual Form of (22)

In this subsection, we derive the Fenchel dual [36] form of (22). To simplify presentation, we use $\mathbf{w}$ to represent $\mathbf{W}_{\cdot i}$. Then we re-formulate the primal form in (22) as:

$$\min_{\mathbf{w}} \mathcal{P}(\mathbf{w}) = \mathcal{M}(\mathbf{w}) + \mathcal{N}(-\mathbf{w}^T\mathbf{K})$$
$$= \frac{1}{2}||\mathbf{w}||_2^2 + \frac{\lambda}{\mu}\max_j(\boldsymbol{\Delta}^T - \mathbf{w}^T\mathbf{K})_j, \quad (23)$$

where we define $\mathcal{M}(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||_2^2$ and $\mathcal{N}(-\mathbf{w}^T\mathbf{K}) = \frac{\lambda}{\mu}\max_j(\boldsymbol{\Delta}^T - \mathbf{w}^T\mathbf{K})_j$.

Before deriving the dual form of (23), we first introduce the definition (Definition 1) and the main properties (Theorem 1 and 2) of Fenchel duality.

Definition 1. *The Fenchel conjugate of function $f(\boldsymbol{x})$ is defined as $f^*(\boldsymbol{\theta}) = \max_{\boldsymbol{x}\in dom(f)}(\langle\boldsymbol{\theta}, \boldsymbol{x}\rangle - f(\boldsymbol{x}))$.*

Theorem 1. *(Fenchel-Young inequality: [5], Proposition 3.3.4) Any points $\boldsymbol{\theta}$ in the domain of function f^* and $\boldsymbol{x}$ in the domain of function f satisfy the inequality:*

$$f(\boldsymbol{x}) + f^*(\boldsymbol{\theta}) \geq \langle\boldsymbol{\theta}, \boldsymbol{x}\rangle \quad (24)$$

The equality holds if and only if $\boldsymbol{\theta} \in \partial f(\boldsymbol{x})$.

Theorem 2. *(Fenchel Duality inequality, see e.g.,Theorem 3.3.5 in [5]) Let $\mathcal{M} : \mathbb{R}^d \to (-\infty, +\infty]$ and $\mathcal{N} : \mathbb{R}^p \to (-\infty, +\infty]$ be two closed and convex functions, and $\mathbf{K}$ be a $\mathbb{R}^{d\times p}$ matrix. Then we have*

$$\sup_{\boldsymbol{\alpha}} -\mathcal{M}^*(\mathbf{K}\boldsymbol{\alpha}) - \mathcal{N}^*(\boldsymbol{\alpha}) \leq \inf_{\boldsymbol{w}} \mathcal{M}(\boldsymbol{w}) + \mathcal{N}(-\boldsymbol{w}^T\mathbf{K}), \quad (25)$$

where $\boldsymbol{\alpha} \in \mathbb{R}^p$ and $\boldsymbol{w} \in \mathbb{R}^d$. The equality holds if and only if $0 \in (dom(\mathcal{N}) - \mathbf{K}^T dom(\mathcal{M}))$.

Note that the right hand side of the inequality in (25) is called the primal form and the left hand side of (25) is the corresponding dual form.

With Definition 1, it is known (see, e.g., [38], Appendix B) that the Fenchel dual norm (i.e., the Fenchel conjugate) of the ℓ_2-norm $f(\mathbf{x}) = \frac{1}{2}||\mathbf{x}||_2^2$ is also the ℓ_2-norm $f^*(\theta) = \frac{1}{2}||\theta||_2^2$. Hence, the Fenchel conjugate of $\mathcal{M}(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||_2^2$ is

$$\mathcal{M}^*(-\mathbf{K}\alpha) = \frac{1}{2}||-\mathbf{K}\alpha||_2^2 \quad (26)$$

It is known ([38], Appendix B) that the Fenchel conjugate of $f(\mathbf{x}+\mathbf{y})$ is $f^*(\theta) - \langle\theta, \mathbf{y}\rangle$, the Fenchel conjugate of $cf(\mathbf{x})$ $(c > 0)$ is $cf^*(\theta/c)$. Then we can derive that the Fenchel conjugate of $cf(\mathbf{x}+\mathbf{y})$ is

$$cf^*(\theta/c) - \langle\theta, \mathbf{y}\rangle. \quad (27)$$

In addition, the Fenchel conjugate of $f(\mathbf{x}) = \max_j(\mathbf{x}_j)$ is $I_{\theta_i\geq 0,\sum_i \theta_i=1}(\theta)$ with $I_{condition}(.)$ being the indicator function that $I_{condition}(\theta) = 0$ if *condition* is true and otherwise $I_{condition}(\theta) = +\infty$ (see [38], Appendix B). For convenience, we denote $\mathcal{Q}(\mathbf{x}) = \max_j(\mathbf{x}_j)$. It is easy to verify that $\mathcal{N}(-\mathbf{w}^T\mathbf{K}) = \frac{\lambda}{\mu}\max_j(\Delta^T - \mathbf{w}^T\mathbf{K})_j = \frac{\lambda}{\mu}\mathcal{Q}(\Delta^T - \mathbf{w}^T\mathbf{K})$. Hence, by using (27), the Fenchel conjugate of $\mathcal{N}(-\mathbf{w}^T\mathbf{K})$ is:

$$\mathcal{N}^*(\alpha) = \frac{\lambda}{\mu}\mathcal{Q}^*((\alpha)/(\frac{\lambda}{\mu})) - \langle\alpha, \boldsymbol{\Delta}\rangle$$
$$= \begin{cases} -\boldsymbol{\Delta}^T\alpha, & \sum_{k=1}^{p}\alpha_k = \frac{\lambda}{\mu} \text{ and } \alpha_k \geq 0, \ k = 1, \cdots, p; \\ +\infty, & otherwise. \end{cases} \quad (28)$$

With (26), (28) and (25), we have that the dual form of (23) is:

$$\max_{\alpha} \mathcal{D}(\alpha)$$
$$= \max_{\alpha} -\mathcal{M}^*(\mathbf{K}\alpha) - \mathcal{N}^*(\alpha)$$
$$= \max_{\alpha} -\frac{1}{2}\alpha^T\mathbf{K}^T\mathbf{K}\alpha + \boldsymbol{\Delta}^T\alpha \quad (29)$$
$$s.t. \sum_{k=1}^{p}\alpha_k = \frac{\lambda}{\mu} \text{ and } \alpha_k \geq 0, \ k = 1, \cdots, p$$

The dual form in (29) is a smooth quadratic function with linear constraints, which is easier to be optimized compared to its primal form in (23).

5.3.3 Primal-Dual Algorithm via Coordinate Ascent

In this subsection, we develop a coordinate ascent algorithm to optimize the objective in (29), where we use the primal-dual gap $\mathcal{P}(\mathbf{w}) - \mathcal{D}(\alpha)$ as the early stopping criterion. Coordinate ascent is a widely-used method in various machine learning problems (e.g., [12], [38], [23], [45]).

Algorithm 2 Primal-dual algorithm via coordinate ascent

Input: $\mathbf{b}_i, \epsilon_F, \lambda, \mu$, maximal iteration number T_F
Output: $\mathbf{w}$
1. Initialize: $v \leftarrow 0$, $\hat{\mathbf{w}} \leftarrow 0$
2. Repeat:
3.   Find the largest element $(g_\alpha)_j$ in the gradient vector $g_\alpha = \nabla\mathcal{D}(\alpha)$ by solving (30) via Algorithm 3 for F-score (or Algorithm 4 for AUC).
4. $\boldsymbol{\Delta}_j \leftarrow \Delta(\mathbf{y}^{(i)}, \mathbf{y}_j)$
5. $\mathbf{K}_{\cdot j} \leftarrow \mathbf{X}^{(i)T}(\mathbf{y}^{(i)} - \mathbf{y}_j) + \frac{\mu}{\lambda}\mathbf{b}_i$
6.   Calculate γ by (37).
7.   Update $\hat{\mathbf{w}}$ by (35).
8.   Update v by (36)
9. Until $\hat{\mathbf{w}}^T\hat{\mathbf{w}} + \max_j(g_\alpha)_j \leq \epsilon_F$ or iteration number reaches T_F
10. $\mathbf{w} \leftarrow \hat{\mathbf{w}}$

The proposed coordinate ascent algorithm is shown in Algorithm 2. Next, we sketch the main steps the proposed algorithm in the following:

Repeat

- Select an index j with the j-th element $(\nabla_\alpha\mathcal{D}(\alpha))_j$ in the gradient vector $\nabla_\alpha\mathcal{D}(\alpha)$ having the largest element.
- Update α_j with other α_k $(k \neq j)$ fixed, in a manner of greedily increasing the value of $\mathcal{D}(\alpha)$.

Until the early stopping criterion $\mathcal{P}(\mathbf{w}) - \mathcal{D}(\alpha) \leq \epsilon_F$ is satisfied.

In each iteration, the proposed algorithm has three main building blocks:

The First Step is to select an index j that the j-th element is the largest element in the gradient vector for the dual objective $\mathcal{D}(\alpha)$. Specifically, the gradient vector with respect to α for $\mathcal{D}(\alpha)$ is:

$$g_\alpha = \nabla_\alpha \mathcal{D}(\alpha) = -\mathbf{K}^T \mathbf{K}\alpha + \boldsymbol{\Delta},$$

and the largest element in $\nabla_\alpha \mathcal{D}(\alpha)$ is:

$$(g_\alpha)_j = (\nabla_\alpha \mathcal{D}(\alpha))_j = \max_j \boldsymbol{\Delta}_j - (\mathbf{K}\alpha)^T \mathbf{K}_{.j}.$$

We denote $\hat{\mathbf{w}} = \mathbf{K}\alpha$. Then, with the definition of $\boldsymbol{\Delta}_j$ and $\mathbf{K}_{.j}$, we have:

$$(\nabla_\alpha \mathcal{D}(\alpha))_j = \max_j \Delta(\mathbf{y}^{(i)}, \mathbf{y}_j) - \hat{\mathbf{w}}^T \mathbf{X}^{(i)T}(\mathbf{y}^{(i)} - \mathbf{y}_j). \tag{30}$$

Interestingly, the problem in (30) is essentially the same as the problems of "finding the most violated constraint" in Structured-SVMs (e.g., the problem (7) in [20]). For several commonly-used evaluation metrics $\Delta(.,.)$, efficient algorithm in polynomial-time were proposed to solve the problems of "finding the most violated constraint". One can directly use these inference algorithms to solve (30) of selecting the largest element from the gradient vector $\nabla_\alpha \mathcal{D}(\alpha)$. For example, when $\Delta(.,.)$ corresponds to F-score, one can use Algorithm 2 in [20] to solve (30); when $\Delta(.,.)$ corresponds to AUC, one can use Algorithm 3 in [20] to solve (30). For self-containness, we shown these two algorithms with our notations in Algorithm 3 and 4. Note that Algorithm 3 and 4 have the time complexity in $O(n_i^2)$ and $O(n_i \log n_i)$, respectively.

Algorithm 3 Algorithm to solve (30) with loss function defined on F-score

---

Input: $n = n_i, \mathbf{X}^{(i)} = (\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_n^{(i)})^T$,
    $\mathbf{y}^{(i)} = (\mathbf{y}_1^{(i)}, \ldots, \mathbf{y}_n^{(i)})^T$, $\mathbf{w}$
Output: $\mathbf{y}_j$
1. Initialize: $(k_1^p, \ldots, k_{Pos}^p) \leftarrow sort\{k : \mathbf{y}_k^{(i)} = 1\}$ by $\mathbf{w}^T \mathbf{x}_k^{(i)}$
    $(k_1^n, \ldots, k_{Neg}^n) \leftarrow sort\{k : \mathbf{y}_k^{(i)} = -1\}$ by $\mathbf{w}^T \mathbf{x}_k^{(i)}$
2. For $a \in [0, \ldots, Pos]$ do:
3. $c \leftarrow Pos - a$
4.   Set $l_{k_1^p}, \ldots, l_{k_a^p}$ to 1 and set $l_{k_{a+1}^p}, \ldots, l_{k_{Pos}^p}$ to -1
5.   For $d \in [0, \ldots, Neg]$ do:
6. $b \leftarrow Neg - d$
7.   Set $l_{k_1^n}, \ldots, l_{k_b^n}$ to 1 and set $l_{k_{b+1}^n}, \ldots, l_{k_{Neg}^n}$ to -1
8. $v \leftarrow \Delta(\mathbf{y}^{(i)}, (l_1, \ldots, l_n)^T) + \mathbf{w}^T \sum_{k=1}^{n} l_k \mathbf{x}_k^{(i)}$,
        where $\Delta(\cdot, \cdot)$ is defined by (11)
9.   If v is the largest so far, then:
10. $\mathbf{y}_j \leftarrow (l_1, \ldots, l_n)^T$
11.   End if
12.   End for
13. End for

The Second Step is to update α_j by fixing other variable $\alpha_k (k \neq j)$, given the selected index j.

We define the update rules for α as:

$$\alpha \leftarrow (1 - \gamma)\alpha + \frac{\gamma\lambda}{\mu} e_j, \tag{31}$$

Algorithm 4 Algorithm to solve (30) with loss function defined on AUC

Input: $n = n_i, \mathbf{X}^{(i)} = (\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_n^{(i)})^T$,
    $\mathbf{y}^{(i)} = (\mathbf{y}_1^{(i)}, \ldots, \mathbf{y}_n^{(i)})^T$, $\mathbf{w}$
Output: $\mathbf{y}_j$
1. Initialize: for $k \in \{k : \mathbf{y}_k^{(i)} = 1\}$ do $q_k \leftarrow -0.25 + \mathbf{w}^T \mathbf{x}_k^{(i)}$
        for $k \in \{k : \mathbf{y}_k^{(i)} = -1\}$ do $q_k \leftarrow 0.25 + \mathbf{w}^T \mathbf{x}_k^{(i)}$
2. $(r_1, \ldots, r_n) \leftarrow$ sort $\{1, \ldots, n\}$ by q_k
3. $q_{Pos} \leftarrow Pos$, $q_{Neg} \leftarrow 0$
4. For $k \in [1, \ldots, n]$ do:
5.   If $\mathbf{y}_{r_k}^{(i)} > 0$, then:
6. $l_{r_k} \leftarrow (Neg - 2q_n)$
7. $q_{Pos} \leftarrow q_{Pos} - 1$
8.   else
9. $l_{r_k} \leftarrow (-Pos + 2q_{Pos})$
10. $q_{Neg} \leftarrow q_{Neg} + 1$
11.   End if
12. End for
13. Convert $(l_1, \ldots, l_n)$ to $\mathbf{y}_j$ according to some threshold value.

---

where $0 \leq \gamma \leq 1$ and e_j denotes the n_i-dimension vector with the j-th element being one and other elements being zeros. It is worth noting that, given $\alpha_j \geq 0$ and $\sum_j \alpha_j = \lambda/\mu$ before updating, and $0 \leq \gamma \leq 1$, this form of rules in (31) guarantees that $\alpha_j \geq 0$ and $\sum_j \alpha_j = \lambda/\mu$ still hold after updating.

By substituting (31) into (29), we obtain the corresponding optimization problem with respect to γ:

$$\max_\gamma -\frac{1}{2}[(1-\gamma)\alpha + \tfrac{\gamma\lambda}{\mu}e_j]^T \mathbf{K}^T \mathbf{K}[(1-\gamma)\alpha + \tfrac{\gamma\lambda}{\mu}e_j]$$
$$+[(1-\gamma)\alpha + \tfrac{\gamma\lambda}{\mu}e_j]^T \boldsymbol{\Delta} \tag{32}$$

Intuitively, our goal is to find $\gamma \in [0, 1]$ to increase the dual objective $\mathcal{D}(\alpha)$ as much as possible. By setting the gradient of (32) with respect to γ to zero, we have

$$||\mathbf{K}(e_j\lambda/\mu - \alpha)||_2^2 \gamma + (e_j\lambda/\mu - \alpha)^T \mathbf{K}^T \mathbf{K}\alpha$$
$$-(e_j\lambda/\mu - \alpha)^T \boldsymbol{\Delta} = 0$$

By simple algebra, we have

$$\gamma = -\frac{(e_j\lambda/\mu - \alpha)^T (\mathbf{K}^T \mathbf{K}\alpha - \boldsymbol{\Delta})}{||\mathbf{K}(e_j\lambda/\mu - \alpha)||_2^2} \tag{33}$$

To ensure that $0 \leq \gamma \leq 1$, we make further restriction on γ:

$$\gamma = \max(\min(-\frac{(e_j\lambda/\mu - \alpha)^T (\mathbf{K}^T \mathbf{K}\alpha - \boldsymbol{\Delta})}{||\mathbf{K}(e_j\lambda/\mu - \alpha)||_2^2}, 1), 0) \tag{34}$$

The calculation of γ in (34) depends on the calculation of $\mathbf{K}\alpha$ and $\alpha^T \boldsymbol{\Delta}$. However, since $\mathbf{K} \in \mathbb{R}^{d \times p}$, $\boldsymbol{\Delta}, \alpha \in \mathbb{R}^p$ and $p = 2^{n_i}$, the time of directly calculating either $\mathbf{K}\alpha$ or $\alpha^T \boldsymbol{\Delta}$ depends exponentially on n_i, which may often unaffordable. In order to improve efficiency, we maintain auxiliary variable to reduce the computation cost. Remind that we have defined $\hat{\mathbf{w}} = \mathbf{K}\alpha$. We also define $v = \alpha^T \boldsymbol{\Delta}$. We maintain $\hat{\mathbf{w}}$ and v during the iterations.

With the update rule (31) for α, we can easily derive the corresponding update rules for $\hat{\mathbf{w}}$ and v, respectively:

$$\hat{\mathbf{w}} \leftarrow (1 - \gamma)\hat{\mathbf{w}} + \frac{\gamma\lambda}{\mu}\mathbf{K}_{\cdot j}, \qquad (35)$$

$$v \leftarrow (1 - \gamma)v + \frac{\gamma\lambda}{\mu}\boldsymbol{\Delta}_j. \qquad (36)$$

Obviously, the update rule for $\hat{\mathbf{w}}$ (or v) has the time complexity $O(d)$ (or $O(1)$).

With the maintained $\hat{\mathbf{w}}$ and v, the update rule in (34) can be simplified to:

$$\gamma \leftarrow \max(\min(-\frac{\frac{\lambda}{\mu}(\mathbf{K}_{\cdot j}^T\hat{\mathbf{w}} - \boldsymbol{\Delta}_j) - \hat{\mathbf{w}}^T\hat{\mathbf{w}} + v}{||\frac{\lambda}{\mu}\mathbf{K}_{\cdot j} - \hat{\mathbf{w}})||_2^2}, 1), 0), \quad (37)$$

where the time complexity of update γ in (37) is reduced to $O(d)$.

The early stopping criterion is defined based on the primal-dual gap $\mathcal{P}(\mathbf{w}) - \mathcal{D}(\alpha) \leq \epsilon_F$ where the parameter ϵ_F is the pre-defined tolerance. Assume $\mathcal{P}(\mathbf{w}^\star)$ is the optimal value of the primal objective (23). According to Theorem 2, we have:

$$\mathcal{P}(\mathbf{w}) - \mathcal{P}(\mathbf{w}^\star) \leq \mathcal{P}(\mathbf{w}) - \mathcal{D}(\alpha) \leq \epsilon_F.$$

It is worth noting that, by using the update rule (31) with $0 \leq \gamma \leq 1$, Algorithm 2 guarantees that α satisfies the constraints $\alpha_k \geq 0$ and $\sum_k \alpha_k = \lambda/\mu$ in all of the iterations. In order words, we have $\mathcal{N}^*(\alpha) < \infty$ in all of the iterations. Hence, with (23) and (29), we have:

$$\begin{aligned} &\mathcal{P}(\mathbf{w}) - \mathcal{D}(\alpha) \\ &= \mathcal{M}(\mathbf{w}) + \mathcal{M}^*(\mathbf{K}\alpha) + \mathcal{N}(-\mathbf{w}^T\mathbf{K}) + \mathcal{N}^*(\alpha) \end{aligned} \qquad (38)$$

With Theorem 1, we have $\mathcal{M}(\mathbf{w}) + \mathcal{M}^*(\mathbf{K}\alpha) \geq \langle \mathbf{w}, \mathbf{K}\alpha \rangle$, where the equality holds when $\mathbf{w} = \mathbf{K}\alpha = \hat{\mathbf{w}}$. In order to greedily upper-bounded the gap $\mathcal{D}(\alpha^\star) - \mathcal{D}(\alpha)$, we set $\mathbf{w} = \mathbf{K}\alpha = \hat{\mathbf{w}}$ in (38) and obtain:

$$\begin{aligned} &\mathcal{P}(\mathbf{w}) - \mathcal{D}(\alpha) \\ &= \langle \hat{\mathbf{w}}, \mathbf{K}\alpha \rangle + \mathcal{N}(\hat{\mathbf{w}}^T\mathbf{K}) + \mathcal{N}^*(\alpha) \\ &= \hat{\mathbf{w}}^T\hat{\mathbf{w}} + \max_j (g_\alpha)_j - v \end{aligned} \qquad (39)$$

Consequently, the early stopping criterion is set to be $\hat{\mathbf{w}}^T\hat{\mathbf{w}} + \max_j(g_\alpha)_j - v \leq \epsilon_F$, which can be calculated in time $O(d)$.

5.4 Convergence Analysis

For the sub-problem w. r. t. $\mathbf{W}$ (see Section 5.3), the proposed coordinate ascent method is similar to those in [38], [23]. By using similar proof techniques to those of [38], [23] (e.g., see the proofs of Theorem 1 in [23]), we can derive that, after T iteration in Algorithm 2, we have $\mathcal{D}(\alpha^\star) - \mathcal{D}(\alpha) \leq \mathcal{P}(\mathbf{w}) - \mathcal{D}(\alpha) \leq \epsilon_F = O(\frac{1}{T})$. Note that $\mathcal{D}(\alpha^\star) = \mathcal{P}(\mathbf{w}^\star)$, where $\mathcal{D}(\alpha^\star)$ and $\mathcal{P}(\mathbf{w}^\star)$ are the optimal solution of (29) and (23) respectively. Ideally, for all the tasks, if we set the iteration number T to be sufficient

large, we can solve the sub-problem w,r.t. W exactly (by ignoring the small numerical errors).

In addition, as discussed in Section 5.2, the sub-problems w. r. t. $\mathbf{S}$ can be solved exactly by closed-form solutions. Hence, the objective (12) is convex subject to linear constraints, and both of its subproblems can be solved exactly. Based on existing theoretical results [6], [16], we have that Algorithm 1 converges to global optima with a $O(1/\epsilon)$ convergence rate.

6 EXPERIMENTS

6.1 Overview

In this section, we evaluate and compare the performance of the proposed SMTL method on several benchmark datasets. For the regularizer $\Omega(\mathbf{S})$ in (12), we consider $||\mathbf{S}||_{1,1}$, $||\mathbf{S}||_{2,1}$ and $||\mathbf{S}||_*$, respectively. For the evaluation metric $\Delta(.,.)$ used in $\mathcal{G}(\mathbf{W})$ in (12), we consider F_1-score (with $\beta = 1$) and AUC. These settings lead to six variants of SMTL.

Here we focus on MTL for classification. Given a specific regularizer (i.e., $||\mathbf{S}||_{1,1}$, $||\mathbf{S}||_{2,1}$ or $||\mathbf{S}||_*$), we choose these methods as baselines: (1) single-task structured SVM that directly optimizes AUC (StructSVM) [20], we train it on each of the individual tasks and average the results. (2) MTL with hinge loss for classification (MTL-CLS). (3) MTL with least squares loss for regression (MTL-REG). (4) RAkEL, a meta algorithm using random k-label sets [41]. (5) MLCSSP, a method spanning the original label space by subset of labels [4]. (6) AdaBoostMH, a method based on AdaBoost [37]. (7) HOMER, a method based on the hierarchy of multi-label learners [40]. (8) BR, the binary relevance method [42]. (9) LP, the label power-set method [42]. (10) ECC, the ensembles of classifier chains method (ECC) [35]. Note that the classification problem can be regarded as a regression problem[2].

The proposed methods, the baselines MTL-CLS and MTL-REG were implemented with Python 2.7. For MTL-REG, our implementations are based on the algorithms in [28] (for the $\ell_{2,1}$ norm) and [17] (for the trace norm). According to Theorem 3 in [20], the problem of MTL-CLS is equivalent to a special form of SMTL in (2) (with $\Delta(y^{(i)}, y) = 2 \times t$, where t represents the number of index k that satisfies $y_k^{(i)} \neq y_k$). Hence, our implementation of MTL-CLS is based on the framework of Algorithm 1. For StructSVM, we use the open-source implementation of SVM-Perf [20]. All the experiments were conducted on a Dell PowerEdge R320 server with 16G memory and 1.9Ghz E5-2420 CPU.

---

2. For a dataset for binary classification that each positive example has a label $+1$ and each negative example has a label -1, one can regard these labels as real numbers (i.e., 1.0 for each of the positive examples and -1.0 for each of the negative examples). Then, this dataset can be used in a MTL method for regression to learn a regressor. After obtaining the regressor, for a test example x, if the predicted label of x (by the regressor) is larger than 0, one can regard x as a positive example. On the other hand, if the predicted label of x is smaller than 0, then one can regard x as a negative example.

We report the experimental results on 9 real-world datasets. The statistics of these datasets are summarized in Table 1. In the Emotions dataset, the labels are 6 kinds of emotions, and the features are rhythmic and timbre extracted from music wave files. In the Yeast dataset, the labels are localization sites of protein, and the features are protein properties. In the Flags dataset, the labels are religions of countries and the features are extracted from flag images. In the Cal500 dataset, the labels are semantically meaning of popular songs and the features are extracted from audio data. In the Segmentation dataset, the labels are content of image region, and the features are pixels' properties of image regions. In the Optdigits dataset, the labels are handwritten digits 0 to 9, and the features are pixels. In the MediaMill dataset, the labels are semantic concepts of each video and the features are extracted from videos. In the TMC2007 dataset, the labels are the document topics, and the features are discrete attributes about terms. In the Scene dataset, the labels are scene types, and the features are spatial color moments in LUV space. All of these datasets are normalized.

TABLE 1: Statistics of 9 datasets

	Type	Features	Samples	Tasks
Emotions	music	72	593	6
Yeast	gene	103	2417	14
Flags	image	19	194	7
Cal500	songs	68	502	174
Segmentation	image	19	2310	7
Optdigits	image	64	5620	10
MediaMill	multimedia	120	10000	12
TMC2007	test	500	10000	6
Scene	image	294	2407	6

Following the settings in [9], to evaluate the performance, we use AUC, Macro F1-score, and Micro F1-score as the evaluation metrics (the details about the computation of AUC and F_1[3] can be found in Section 4).

For each dataset, we firstly generate 10 60%:40% partitions. In each partition, the "60%" part is used as the training set and the "40%" part is used as the test set. Then, we run each of the methods (the baselines and the proposed methods) on these 10 partitions, and reported the averaged results on these 10 trials. Note that, for a fair comparison, in a dataset, each method uses the same ten partitions to produce its results. After the training set is determined, we conduct 10-fold cross validation on the

3. In MTL, the Macro F_1 is calculated by firstly calculating the F_1 score of each individual task, and then average these F_1 scores over all tasks. The Micro F_1 in MTL is calculated by $\frac{2 \times P \times R}{P+R}$, where

$$P = \frac{\sum_{i=1}^{m} \sum_{k=1}^{n_i} I(\mathbf{y}_k^{(i)} = 1 \text{ and } (\mathbf{y}_j)_k = 1)}{\sum_{i=1}^{m} \sum_{k=1}^{n_i} I(\mathbf{y}_k^{(i)} = 1)},$$

$$R = \frac{\sum_{i=1}^{m} \sum_{k=1}^{n_i} I(\mathbf{y}_k^{(i)} = 1 \text{ and } (\mathbf{y}_j)_k = 1)}{\sum_{i=1}^{m} \sum_{k=1}^{n_i} I((\mathbf{y}_j)_k = 1)}.$$

TABLE 2: Comparison results on Cal500, Segmentation and Optdigits.

METHOD	MACRO F_1	MICRO F_1	Average AUC
Cal500			
SMTL($\ell_{2,1}$+AUC)	**21.722±0.456**	38.452±0.610	**56.505±0.511**
SMTL($\ell_{2,1}$+F_1)	21.495±0.232	**40.127±0.173**	53.690±0.293
MTL-CLS($\ell_{2,1}$)	13.157±0.449	37.357±0.180	55.764±0.820
MTL-REG($\ell_{2,1}$)	12.500±0.129	36.438±0.176	52.964±0.758
SMTL($\ell_{1,1}$+AUC)	**21.721±0.807**	35.52±0.811	**56.716±0.500**
SMTL($\ell_{1,1}$+F_1)	21.138±0.191	**38.386±0.456**	53.358±0.827
MTL-CLS($\ell_{1,1}$)	12.176±0.445	37.387±0.845	56.316±0.216
MTL-REG($\ell_{1,1}$)	12.447±0.297	36.66±0.638	53.628±0.264
SMTL(TraceNorm+AUC)	**21.772±0.545**	35.204±0.585	**56.798±0.358**
SMTL(TraceNorm+F_1)	21.768±0.333	**38.559±0.394**	54.987±0.823
MTL-CLS(TraceNorm)	12.884±0.353	37.402±0.501	55.635±0.511
MTL-REG(TraceNorm)	8.348±0.999	34.832±0.698	55.69±0.636
StructSVM	20.864±1.150	35.408±1.150	51.427±0.841
RAkEL	20.628±0.611	33.689±0.843	54.637±0.656
MLCSSP	21.677±0.514	27.093±0.537	52.69±0.983
AdaBoostMH	0.923±0.274	6.492±0.146	50.734±0.538
HOMER	13.850±0.163	30.332±1.313	52.461±0.937
BR	17.094±0.634	33.619±0.375	50.563±0.153
LP	15.257±0.428	32.978±0.668	52.117±0.685
ECC	9.600±0.666	34.789±0.482	52.117±0.625
Segmentation			
SMTL($\ell_{2,1}$+AUC)	72.832±1.567	68.445±1.543	**97.195±0.4549**
SMTL($\ell_{2,1}$+F_1)	**85.61±1.304**	84.149±1.684	96.967±0.647
MTL-CLS($\ell_{2,1}$)	85.114±1.946	**84.228±4.508**	96.93±0.560
MTL-REG($\ell_{2,1}$)	75.547±1.215	81.702±2.456	96.757±0.645
SMTL($\ell_{1,1}$+AUC)	73.378±1.564	68.424±1.787	**97.527±0.286**
SMTL($\ell_{1,1}$+F_1)	**85.105±1.830**	**83.693±1.192**	96.757±0.192
MTL-CLS($\ell_{1,1}$)	83.712±3.513	82.518±4.003	96.781±0.828
MTL-REG($\ell_{1,1}$)	76.253±2.564	82.606±0.156	96.798±0.231
SMTL(TraceNorm+AUC)	72.265±1.453	67.655±1.978	**97.134±0.457**
SMTL(TraceNorm+F_1)	**85.356±1.092**	**83.462±1.805**	96.863±0.322
MTL-CLS(TraceNorm)	82.703±3.865	82.150±5.439	96.705±0.612
MTL-REG(TraceNorm)	76.602±1.286	82.805±1.877	96.698±0.147
StructSVM	44.632±1.828	53.992±1.828	89.355±0.311
RAkEL	75.592±0.243	70.980±0.398	91.333±0.082
MLCSSP	79.821±8.533	78.923±14.036	93.810±0.329
AdaBoostMH	75.633±0.209	71.018±0.376	96.148±0.089
HOMER	72.920±2.505	69.969±1.651	91.225±1.543
BR	84.236±0.638	78.796±0.708	96.870±0.194
LP	84.394±0.603	83.411±0.615	96.240±0.124
ECC	84.183±0.550	82.942±0.542	96.782±0.269
Optdigits			
SMTL($\ell_{2,1}$+AUC)	92.722±0.595	92.734±0.712	**99.657±0.0528**
SMTL($\ell_{2,1}$+F_1)	**93.963±0.164**	**93.964±0.235**	99.589±0.054
MTL-CLS($\ell_{2,1}$)	93.701±0.403	92.773±0.440	99.206±0.044
MTL-REG($\ell_{2,1}$)	88.901±0.306	89.268±0.875	99.32±0.089
SMTL($\ell_{1,1}$+AUC)	92.526±0.624	92.213±0.670	**99.653±0.078**
SMTL($\ell_{1,1}$+F_1)	**93.692±0.508**	**94.626±0.520**	99.554±0.047
MTL-CLS($\ell_{1,1}$)	92.961±0.608	94.009±0.356	98.658±0.067
MTL-REG($\ell_{1,1}$)	88.762±0.845	89.203±0.865	99.269±0.045
SMTL(TraceNorm+AUC)	92.862±0.543	92.802±0.944	**99.654±0.036**
SMTL(TraceNorm+F_1)	**94.206±0.202**	**94.139±0.266**	99.566±0.027
MTL-CLS(TraceNorm)	93.701±0.435	93.773±0.267	99.182±0.065
MTL-REG(TraceNorm)	88.777±0.765	89.173±0.946	99.293±0.048
StructSVM	36.276±0.905	38.289±2.218	98.400±0.366
RAkEL	82.450±0.168	80.967±0.311	94.543±0.070
MLCSSP	75.191±2.245	82.129±3.195	88.879±0.195
AdaBoostMH	93.083±0.695	93.108±0.669	98.594±0.119
HOMER	74.869±4.151	75.713±3.663	93.391±0.964
BR	92.625±0.348	92.714±0.383	99.370±0.122
LP	88.875±0.212	88.915±0.269	94.941±0.329
ECC	93.043±0.206	94.019±0.213	99.019±0.156

training set to choose the trade-off parameter λ within $\{10^{-3} \times i\}_{i=1}^{10} \cup \{10^{-2} \times i\}_{i=1}^{10} \cup \{10^{-1} \times i\}_{i=1}^{10} \cup \{2 \times i\}_{i=1}^{10} \cup \{40 \times i\}_{i=1}^{20}$.

In Algorithm 2, we set the maximum iterations $T_F = 5000$ and the optimization tolerance $\epsilon_F = 10^{-5}$.

6.2 Results on real-world datasets

The evaluation results w.r.t. Micro F_1, Macro F_1 and AUC (with standard deviations) of the proposed SMTL are shown in Table 2, 3 and 4. As can be seen, by using the same regularizer, the proposed SMTL variants that optimize F_1-score or AUC show superior performance gains over the baselines. In most cases, the SMTL variant that optimizes a specific metric achieves the best results on this metric. Here are some statistics.

TABLE 3: Comparison results on Scene, MediaMill and TMC2007.

METHOD	MACRO F_1	MICRO F_1	Average AUC
Scene			
SMTL($\ell_{2,1}$+AUC)	54.013±1.124	54.746±1.231	**89.99±0.820**
SMTL($\ell_{2,1}+F_1$)	**55.787±0.756**	**56.434±0.567**	87.652±0.280
MTL-CLS($\ell_{2,1}$)	54.722±1.590	54.508±1.176	86.738±1.102
MTL-REG($\ell_{2,1}$)	51.157±0.343	52.810±0.345	85.194±0.712
SMTL($\ell_{1,1}$+AUC)	54.296±0.977	54.333±0.025	**88.358±0.467**
SMTL($\ell_{1,1}+F_1$)	**55.501±1.92**	**56.007±2.34**	87.364±1.801
MTL-CLS($\ell_{1,1}$)	54.387±0.730	54.805±1.488	85.952±1.116
MTL-REG($\ell_{1,1}$)	50.748±0.546	51.280±0.619	85.032±0.779
SMTL(TraceNorm+AUC)	54.227±0.660	55.384±0.804	**88.421±1.103**
SMTL(TraceNorm+F_1)	**55.396±1.089**	**56.304±1.119**	87.071±0.682
MTL-CLS(TraceNorm)	55.104±0.298	55.481±0.506	86.205±0.471
MTL-REG(TraceNorm)	50.832±0.226	51.236±0.264	85.275±0.852
StructSVM	49.826±0.815	49.951±0.755	82.375±0.393
RAkEL	54.592±0.613	55.719±0.565	78.981±0.535
MLCSSP	42.764±0.080	47.178±0.181	65.830±2.240
AdaBoostMH	36.506±0.404	40.681±0.449	87.617±0.470
HOMER	60.980±2.470	58.251±2.592	80.744±0.360
BR	54.579±1.813	55.019±1.843	82.888±1.164
LP	54.902±1.503	55.818±1.595	75.900±1.362
ECC	55.347±0.893	55.831±0.881	88.153±0.298
MediaMill			
SMTL($\ell_{2,1}$+AUC)	18.030±0.294	22.058±0.257	66.068±0.426
SMTL($\ell_{2,1}+F_1$)	**22.851±5.093**	**56.424±2.761**	**78.705±2.280**
MTL-CLS($\ell_{2,1}$)	10.613±1.733	55.441±3.647	76.216±2.474
MTL-REG($\ell_{2,1}$)	6.366±0.065	55.515±0.465	53.867±0.496
SMTL($\ell_{1,1}$+AUC)	18.012±0.286	22.232±0.211	65.405±0.503
SMTL($\ell_{1,1}+F_1$)	**22.386±5.326**	**56.169±2.436**	**78.907±1.854**
MTL-CLS($\ell_{1,1}$)	8.542±1.672	55.838±2.229	74.037±1.219
MTL-REG($\ell_{1,1}$)	6.393±0.033	55.687±0.439	53.036±0.181
SMTL(TraceNorm+AUC)	18.201±0.221	22.684±0.354	66.847±1.015
SMTL(TraceNorm+F_1)	**27.973±3.006**	**56.031±4.924**	**79.730±1.850**
MTL-CLS(TraceNorm)	15.800±0.589	50.098±5.569	75.968±2.144
MTL-REG(TraceNorm)	6.380±0.045	55.333±0.425	53.825±0.493
StructSVM	17.847±0.318	22.030±0.284	64.761±0.487
RAkEL	19.874±0.156	26.686±0.189	63.241±0.398
MLCSSP	15.129±0.633	20.124±0.723	52.473±1.884
AdaBoostMH	17.939±0.469	41.991±0.425	61.914±0.167
HOMER	17.939±0.469	41.991±0.425	61.914±0.167
BR	19.769±0.196	26.515±0.166	69.032±0.854
LP	24.135±0.959	50.170±0.402	60.597±0.502
ECC	24.879±0.590	56.214±0.363	78.067±0.705
TMC2007			
SMTL($\ell_{2,1}$+AUC)	59.432±0.581	68.02±1.042	90.138±0.17
SMTL($\ell_{2,1}+F_1$)	**64.321±0.955**	**74.159±0.255**	**90.561±0.669**
MTL-CLS($\ell_{2,1}$)	60.517±1.363	71.284±0.387	88.382±0.398
MTL-REG($\ell_{2,1}$)	37.106±0.416	70.181±0.221	85.218±0.529
SMTL($\ell_{1,1}$+AUC)	60.249±0.147	67.654±0.234	**90.441±0.077**
SMTL($\ell_{1,1}+F_1$)	**65.436±1.239**	**73.984±0.533**	90.238±0.732
MTL-CLS($\ell_{1,1}$)	62.919±0.802	72.745±0.464	89.074±0.59
MTL-REG($\ell_{1,1}$)	37.709±0.32	70.431±0.414	86.612±0.592
SMTL(TraceNorm+AUC)	58.595±0.148	68.056±0.45	88.325±0.182
SMTL(TraceNorm+F_1)	**61.867±1.014**	**72.588±0.350**	**89.328±0.815**
MTL-CLS(TraceNorm)	59.752±0.951	71.863±0.628	87.933±0.428
MTL-REG(TraceNorm)	36.64±0.314	70.118±0.437	84.54±0.743
StructSVM	37.19±0.652	45.027±0.601	88.072±0.289
RAkEL	57.331±0.592	69.813±0.179	81.994±0.134
MLCSSP	56.717±0.790	60.417±1.665	75.246±1.093
AdaBoostMH	15.170±1.893	56.004±1.103	61.466±0.206
HOMER	61.144±0.238	71.429±0.104	84.998±0.589
BR	51.939±1.225	67.873±0.374	84.616±0.528
LP	52.683±0.832	62.672±0.526	73.063±0.637
ECC	58.368±0.714	68.223±0.096	86.287±0.664

TABLE 4: Comparison results Emotions, Yeast and Flags.

METHOD	MACRO F_1	MICRO F_1	Average AUC
Emotions			
SMTL($\ell_{2,1}$+AUC)	65.498±2.047	67.067±1.956	**83.378±0.466**
SMTL($\ell_{2,1}+F_1$)	**66.244±1.584**	66.358±1.255	81.986±0.495
MTL-CLS($\ell_{2,1}$)	63.343±1.688	65.684±1.327	80.065±0.490
MTL-REG($\ell_{2,1}$)	62.621±1.543	63.701±1.054	81.32±0.396
SMTL($\ell_{1,1}$+AUC)	65.622±1.984	67.143±1.629	**83.358±0.345**
SMTL($\ell_{1,1}+F_1$)	**67.696±0.348**	**67.923±0.578**	83.106±0.596
MTL-CLS($\ell_{1,1}$)	64.969±0.822	66.584±1.049	80.03±0.574
MTL-REG($\ell_{1,1}$)	62.976±0.547	64.404±1.535	81.811±0.587
SMTL(TraceNorm+AUC)	65.902±1.904	67.405±1.848	**83.362±0.618**
SMTL(TraceNorm+F_1)	**67.600±0.574**	**67.858±0.984**	83.000±0.236
MTL-CLS(TraceNorm)	63.805±2.339	66.602±2.063	80.485±0.597
MTL-REG(TraceNorm)	63.243±1.574	64.869±2.574	82.834±0.266
StructSVM	46.367±5.531	49.902±19.032	62.908±4.361
RAkEL	64.998±1.387	65.835±1.136	75.206±0.875
MLCSSP	62.980±2.780	63.593±2.603	76.054±2.495
AdaBoostMH	4.291±1.429	7.577±2.627	55.111±0.328
HOMER	59.039±2.431	61.830±1.642	71.212±1.167
BR	61.358±2.578	62.635±2.332	79.146±1.250
LP	53.384±1.858	54.618±1.543	68.506±0.652
ECC	62.694±1.645	64.138±1.216	82.589±1.131
Yeast			
SMTL($\ell_{2,1}$+AUC)	43.593±1.120	46.261±0.872	**63.018±1.504**
SMTL($\ell_{2,1}+F_1$)	**44.353±1.080**	**55.451±0.457**	61.285±1.246
MTL-CLS($\ell_{2,1}$)	36.308±0.974	43.908±0.499	56.686±0.539
MTL-REG($\ell_{2,1}$)	28.187±1.544	47.029±0.645	62.757±1.745
SMTL($\ell_{1,1}$+AUC)	43.132±1.349	45.729±1.643	**62.626±1.709**
SMTL($\ell_{1,1}+F_1$)	**44.647±1.058**	**54.971±1.187**	61.569±1.945
MTL-CLS($\ell_{1,1}$)	36.89±0.699	44.620±0.553	58.221±0.424
MTL-REG($\ell_{1,1}$)	33.720±1.634	54.682±1.846	50.050±1.563
SMTL(TraceNorm+AUC)	43.58±1.046	46.395±1.067	**63.058±0.634**
SMTL(TraceNorm+F_1)	**44.972±0.765**	**50.431±0.968**	61.819±0.395
MTL-CLS(TraceNorm)	42.275±1.006	44.542±0.460	61.528±0.590
MTL-REG(TraceNorm)	28.178±1.043	47.046±0.126	62.920±0.326
StructSVM	42.669± 2.48	46.298±2.048	61.894±2.488
RAkEL	44.101±0.389	46.086±0.450	61.971±0.753
MLCSSP	41.511±0.837	46.200±1.272	50.756±0.451
AdaBoostMH	12.255±0.041	48.144±0.315	50.805±0.050
HOMER	40.054±1.063	53.745±0.867	62.311±1.265
BR	39.209±0.891	54.153±0.543	62.375±0.408
LP	37.029±0.584	53.059±0.514	56.616±1.394
ECC	37.523±0.310	54.632±0.325	62.105±0.627
Flags			
SMTL($\ell_{2,1}$+AUC)	60.473±1.951	61.666±2.226	73.875±2.563
SMTL($\ell_{2,1}+F_1$)	**70.279±1.744**	**75.047±0.945**	**75.000±0.745**
MTL-CLS($\ell_{2,1}$)	65.233±1.930	71.709±0.955	72.928±1.479
MTL-REG($\ell_{2,1}$)	66.073±0.276	73.005±1.307	71.429±1.105
SMTL($\ell_{1,1}$+AUC)	60.187±1.971	61.618±1.714	74.136±2.805
SMTL($\ell_{1,1}+F_1$)	**69.122±1.975**	**74.259±1.378**	**74.168±1.513**
MTL-CLS($\ell_{1,1}$)	65.532±1.210	72.666±1.752	72.725±0.497
MTL-REG($\ell_{1,1}$)	65.256±0.739	72.246±0.928	71.299±0.998
SMTL(TraceNorm+AUC)	61.435±1.616	62.84±1.481	**74.367±2.373**
SMTL(TraceNorm+F_1)	**68.704±1.650**	**73.132±1.891**	73.145±1.973
MTL-CLS(TraceNorm)	65.236±3.507	72.888±2.156	73.307±2.155
MTL-REG(TraceNorm)	65.257±2.647	72.437±1.918	71.495±0.783
StructSVM	55.683±5.777	51.957±2.048	72.178±3.604
RAkEL	66.496±5.216	64.749±4.688	61.260±3.805
MLCSSP	59.629±1.619	63.215±1.326	55.865±1.909
AdaBoostMH	56.457±4.288	71.268±1.400	69.329±2.043
HOMER	59.018±1.269	63.855±2.259	64.826±0.569
BR	59.421±2.163	67.287±1.876	66.823±2.860
LP	61.801±3.822	69.132±3.200	60.540±4.149
ECC	64.936±3.023	72.715±1.675	73.913±2.339

On the Yeast dataset, the value of Macro F_1 using SMTL($\ell_{2,1}+F_1$) is 44.353%, a 22.16% relative increase compared to the best MTL baseline MTL-CLS($\ell_{2,1}$); the value of Micro F_1 using SMTL($\ell_{2,1}+F_1$) is 55.451%, a 17.91% relative increase compared to the best MTL baseline MTL-REG($\ell_{2,1}$); the value of averaged AUC using SMTL($\ell_{1,1}$+AUC) is 62.626%, a 7.57% relative increase compared to the best MTL baseline MTL-CLS($\ell_{1,1}$). On the Emotions dataset, the proposed SMTL($\ell_{2,1}+F_1$) performs 66.244% at Macro F1, a 4.58% relative increase compared to the best MTL baseline MTL-CLS($\ell_{2,1}$); SMTL($\ell_{2,1}+F_1$) performs 83.378% at AUC, a 2.53% relative increase compared to the best MTL baseline MTL-CLS($\ell_{2,1}$); SMTL(TraceNorm+F_1) performs 67.6% at Macro F1, a 5.95% relative increase compared to the

best MTL baseline MTL-CLS(TraceNorm). On the Cal500 dataset, SMTL($\ell_{1,1}$+AUC) performs 21.721% at Macro F_1, compared to 12.447% of MTL-REG($\ell_{1,1}$, which indicates a 74.51% relative increase; SMTL($\ell_{2,1}+F_1$) performs 40.127% at Micro F_1, compared to 37.357% of MTL-CLS($\ell_{2,1}$, which indicates a 7.41% relative increase.

In addition, we conduct t-tests and Wilcoxon's signed rank test [43] on 9 datasets to investigate whether the improvements of SMTL methods against the baselines are statistically significant. The p-values of t-tests are showed in Table 5 and 6. The p-values of Wilcoxon's tests are showed in Table 7 and 8. As can be seen, most of the p-values are smaller than 0.05, which indicate that the improvements are statistically significant. These results verify the effectiveness of directly optimizing evaluation

Fig. 1: Comparison results on Segmentation, Emotions and Optdigits w.r.t. AUC.

metric in MTL problems.

TABLE 5: t-test: p-values of SMTL against the baselines

Two methods for comparison	Optdigits	TMC2007	MediaMill	Segmentation
Average AUC				
$\ell_{2,1}$: SMTL(AUC) vs. MTL-CLS	4.86E-07	1.49E-13	2.12E-02	4.74E-03
$\ell_{2,1}$: SMTL(AUC) vs. MTL-REG	2.70E-12	1.44E-18	6.58E-01	4.30E-03
Trace: SMTL(AUC) vs. MTL-CLS	6.88E-03	5.20E-03	2.12E-02	4.74E-03
Trace: SMTL(AUC) vs. MTL-REG	3.85E-12	4.59E-11	6.61E-01	4.25E-03
$\ell_{1,1}$: SMTL(AUC) vs. MTL-CLS	5.71E-03	2.95E-05	5.52E-03	4.75E-03
$\ell_{1,1}$: SMTL(AUC) vs. MTL-REG	1.46E-12	1.92E-12	1.57E-08	4.35E-03
Trace: SMTL(AUC) vs. RAkEL	1.87E-26	1.50E-14	2.79E-13	3.27E-14
Trace: SMTL(AUC) vs. MLCSSP	6.02E-10	1.05E-14	9.63E-15	3.24E-01
Trace: SMTL(AUC) vs. AdaBoostMH	2.36E-04	5.42E-20	4.44E-08	3.05E-14
Trace: SMTL(AUC) vs. HOMER	5.23E-12	6.97E-09	4.65E-08	1.04E-12
Trace: SMTL(AUC) vs. BR	2.91E-10	1.31E-16	2.43E-13	5.14E-07
Trace: SMTL(AUC) vs. LP	6.82E-22	1.41E-20	1.44E-03	9.30E-01
Trace: SMTL(AUC) vs. ECC	8.16E-03	9.67E-19	7.52E-03	6.19E-03
Micro F_1				
$\ell_{2,1}$: SMTL(F_1) vs. MTL-CLS	8.37E-02	8.37E-02	3.98E-10	4.89E-02
$\ell_{2,1}$: SMTL(F_1) vs. MTL-REG	3.28E-20	3.28E-20	2.54E-18	1.24E-02
Trace: SMTL(F_1) vs. MTL-CLS	4.68E-03	4.68E-03	4.00E-10	4.96E-02
Trace: SMTL(F_1) vs. MTL-REG	3.01E-14	3.01E-14	2.30E-18	4.92E-03
$\ell_{1,1}$: SMTL(F_1) vs. MTL-CLS	9.54E-03	9.54E-03	4.19E-10	4.75E-01
$\ell_{1,1}$: SMTL(F_1) vs. MTL-REG	6.16E-12	6.16E-12	2.56E-18	1.03E-01
Trace: SMTL(F_1) vs. RAkEL	7.30E-33	4.64E-25	4.93E-09	9.28E-19
Trace: SMTL(F_1) vs. MLCSSP	2.28E-30	1.90E-18	3.28E-14	2.90E-13
Trace: SMTL(F_1) vs. AdaBoostMH	4.53E-16	9.97E-35	9.38E-12	3.65E-06
Trace: SMTL(F_1) vs. HOMER	5.37E-14	1.13E-12	9.68E-12	8.31E-10
Trace: SMTL(F_1) vs. BR	1.61E-06	3.09E-14	5.20E-05	9.79E-03
Trace: SMTL(F_1) vs. LP	3.94E-20	9.73E-24	1.06E-12	1.39E-05
Trace: SMTL(F_1) vs. ECC	3.99E-07	2.76E-08	1.75E-16	1.45E-01
Macro F_1				
$\ell_{2,1}$: SMTL(F_1) vs. MTL-CLS	4.09E-21	1.61E-10	3.98E-10	4.09E-02
$\ell_{2,1}$: SMTL(F_1) vs. MTL-REG	1.47E-26	3.09E-16	2.54E-18	2.98E-12
Trace: SMTL(F_1) vs. MTL-CLS	1.04E-21	1.82E-02	4.00E-10	4.13E-02
Trace: SMTL(F_1) vs. MTL-REG	3.85E-19	5.87E-12	2.30E-18	3.19E-12
$\ell_{1,1}$: SMTL(F_1) vs. MTL-CLS	7.04E-22	9.26E-07	4.19E-10	4.04E-02
$\ell_{1,1}$: SMTL(F_1) vs. MTL-REG	1.94E-24	8.74E-14	2.56E-18	2.57E-12
Trace: SMTL(F_1) vs. RAkEL	6.35E-29	3.99E-10	4.93E-09	3.08E-16
Trace: SMTL(F_1) vs. MLCSSP	6.50E-16	2.29E-10	3.28E-14	4.68E-02
Trace: SMTL(F_1) vs. AdaBoostMH	1.21E-04	2.99E-23	9.38E-12	3.17E-16
Trace: SMTL(F_1) vs. HOMER	1.78E-11	4.33E-02	9.68E-12	2.55E-11
Trace: SMTL(F_1) vs. BR	1.28E-08	1.12E-13	5.20E-05	1.19E-02
Trace: SMTL(F_1) vs. LP	1.67E-19	1.52E-14	1.06E-12	7.49E-01
Trace: SMTL(F_1) vs. ECC	2.83E-01	4.46E-08	1.75E-16	9.52E-01

6.3 Results on imbalanced data

In the scenarios of learning classifiers on imbalanced data (e.g., the number of positive training samples is much less than that of negative training samples), the

TABLE 6: t-test: p-values of SMTL against the baselines

Two methods for comparison	Cal500	Yeast	Emotions	Scene	Flags
Average AUC					
$\ell_{2,1}$: SMTL(AUC) vs. MTL-CLS	2.62E-01	1.02E-12	2.62E-01	4.92E-02	4.30E-02
$\ell_{2,1}$: SMTL(AUC) vs. MTL-REG	7.48E-05	1.47E-09	7.48E-05	4.74E-11	4.21E-02
Trace: SMTL(AUC) vs. MTL-CLS	1.01E-01	8.97E-13	1.01E-01	4.48E-02	4.21E-02
Trace: SMTL(AUC) vs. MTL-REG	3.04E-03	1.53E-09	3.04E-03	5.00E-11	4.37E-02
$\ell_{1,1}$: SMTL(AUC) vs. MTL-CLS	2.18E-03	1.00E-12	2.18E-03	4.56E-02	4.27E-02
$\ell_{1,1}$: SMTL(AUC) vs. MTL-REG	2.55E-06	1.71E-09	2.55E-06	4.67E-11	4.28E-02
Trace: SMTL(AUC) vs. RAkEL	2.62E-12	1.65E-10	4.55E-04	1.48E-01	5.58E-05
Trace: SMTL(AUC) vs. MLCSSP	1.49E-21	1.05E-07	1.22E-04	1.60E-15	6.81E-11
Trace: SMTL(AUC) vs. AdaBoostMH	4.10E-33	1.03E-06	3.61E-23	3.21E-19	2.12E-02
Trace: SMTL(AUC) vs. HOMER	2.30E-13	2.54E-04	8.57E-09	4.33E-02	9.89E-09
Trace: SMTL(AUC) vs. BR	2.23E-16	7.57E-10	3.84E-06	7.92E-02	1.63E-06
Trace: SMTL(AUC) vs. LP	1.12E-14	6.42E-07	9.15E-15	4.97E-01	3.19E-03
Trace: SMTL(AUC) vs. ECC	2.00E-13	2.09E-12	6.45E-07	3.28E-01	9.78E-01
Micro F_1					
$\ell_{2,1}$: SMTL(F_1) vs. MTL-CLS	4.09E-21	2.70E-05	2.62E-01	1.64E-05	3.14E-02
$\ell_{2,1}$: SMTL(F_1) vs. MTL-REG	1.47E-26	5.00E-02	7.48E-05	1.09E-06	1.87E-03
Trace: SMTL(F_1) vs. MTL-CLS	1.04E-21	3.45E-05	1.01E-01	1.42E-05	3.10E-02
Trace: SMTL(F_1) vs. MTL-REG	3.85E-19	4.39E-02	3.04E-03	1.19E-06	1.76E-03
$\ell_{1,1}$: SMTL(F_1) vs. MTL-CLS	7.04E-22	2.16E-05	2.18E-03	1.54E-05	3.13E-02
$\ell_{1,1}$: SMTL(F_1) vs. MTL-REG	1.94E-24	4.21E-02	2.55E-06	1.35E-06	1.87E-03
Trace: SMTL(F_1) vs. RAkEL	4.16E-08	2.54E-03	4.55E-04	3.26E-15	2.85E-08
Trace: SMTL(F_1) vs. MLCSSP	2.82E-10	9.31E-21	1.22E-04	1.80E-16	2.01E-13
Trace: SMTL(F_1) vs. AdaBoostMH	8.68E-17	3.36E-22	3.61E-23	4.76E-02	7.77E-05
Trace: SMTL(F_1) vs. HOMER	5.69E-11	1.08E-01	8.57E-09	4.53E-14	3.25E-10
Trace: SMTL(F_1) vs. BR	7.35E-21	1.08E-01	3.84E-06	2.51E-09	4.96E-06
Trace: SMTL(F_1) vs. LP	1.64E-13	9.74E-11	9.15E-15	1.25E-14	3.39E-08
Trace: SMTL(F_1) vs. ECC	6.21E-14	5.30E-01	6.45E-07	1.05E-02	9.09E-01
Macro F_1					
$\ell_{2,1}$: SMTL(F_1) vs. MTL-CLS	2.43E-02	2.51E-06	7.55E-12	4.09E-02	1.12E-02
$\ell_{2,1}$: SMTL(F_1) vs. MTL-REG	4.24E-10	2.71E-19	2.83E-09	1.40E-10	2.58E-03
Trace: SMTL(F_1) vs. MTL-CLS	1.77E-05	2.34E-06	3.53E-09	4.37E-02	1.13E-02
Trace: SMTL(F_1) vs. MTL-REG	1.43E-04	3.33E-19	2.69E-02	1.55E-10	2.67E-03
$\ell_{1,1}$: SMTL(F_1) vs. MTL-CLS	3.99E-02	2.66E-06	6.38E-12	4.43E-02	1.11E-02
$\ell_{1,1}$: SMTL(F_1) vs. MTL-REG	1.01E-12	2.76E-19	1.09E-06	1.32E-10	2.54E-03
Trace: SMTL(F_1) vs. RAkEL	5.45E-05	5.52E-03	3.24E-05	5.68E-02	2.11E-04
Trace: SMTL(F_1) vs. MLCSSP	6.64E-01	1.57E-08	6.89E-05	2.84E-02	2.75E-10
Trace: SMTL(F_1) vs. AdaBoostMH	1.28E-29	1.36E-28	3.03E-28	5.89E-21	1.16E-07
Trace: SMTL(F_1) vs. HOMER	3.52E-23	6.16E-10	2.60E-09	3.77E-06	1.84E-11
Trace: SMTL(F_1) vs. BR	7.28E-14	6.97E-12	6.40E-07	2.49E-01	2.86E-09
Trace: SMTL(F_1) vs. LP	1.42E-18	9.45E-16	7.63E-15	4.04E-01	5.68E-05
Trace: SMTL(F_1) vs. ECC	5.69E-21	1.98E-16	5.51E-08	9.15E-01	4.96E-03

TABLE 7: Wilcoxon's test: p-values of SMTL against the baselines

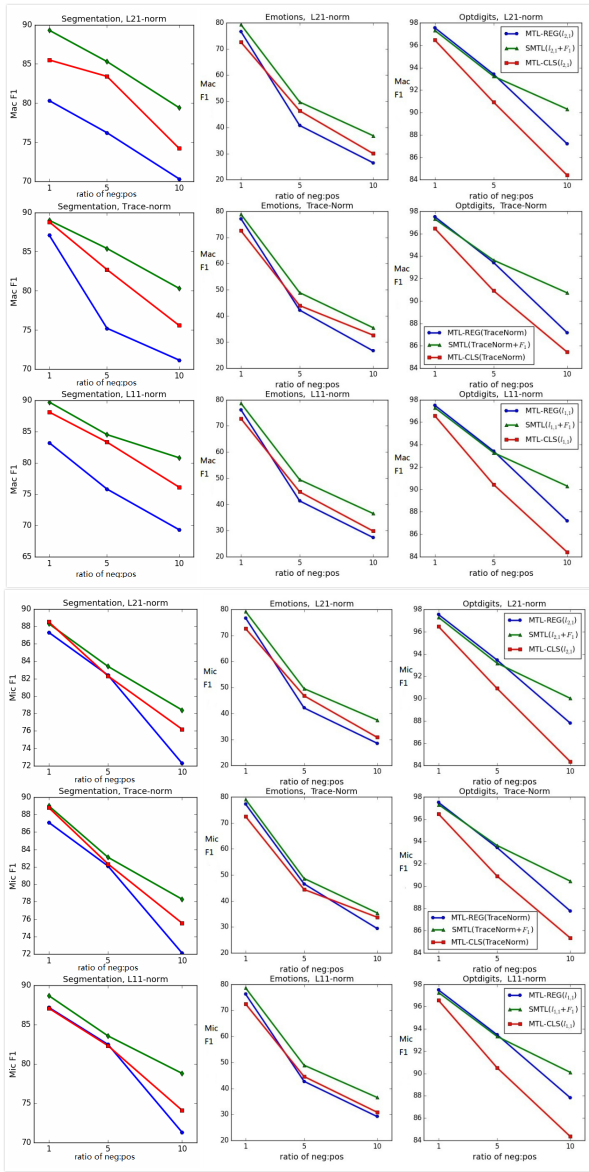| Two methods for comparison	Optdigits	TMC2007	MediaMill	Segmentation
Average AUC				
$\ell_{2,1}$: SMTL(AUC) vs. MTL-CLS	1.25E-02	1.25E-02	5.06E-03	5.75E-01
$\ell_{2,1}$: SMTL(AUC) vs. MTL-REG	5.06E-03	5.06E-03	4.45E-01	2.84E-02
Trace: SMTL(AUC) vs. MTL-CLS	2.84E-02	2.18E-02	2.18E-02	4.69E-02
Trace: SMTL(AUC) vs. MTL-REG	5.06E-03	5.06E-03	8.79E-01	3.86E-01
$\ell_{1,1}$: SMTL(AUC) vs. MTL-CLS	2.84E-02	2.84E-02	4.69E-02	2.18E-02
$\ell_{1,1}$: SMTL(AUC) vs. MTL-REG	5.06E-03	5.06E-03	5.75E-01	2.18E-02
Trace: SMTL(AUC) vs. RAkEL	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(AUC) vs. MLCSSP	5.06E-03	5.06E-03	5.06E-03	2.85E-01
Trace: SMTL(AUC) vs. AdaBoostMH	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(AUC) vs. HOMER	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(AUC) vs. BR	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(AUC) vs. LP	5.06E-03	5.06E-03	1.25E-02	8.79E-01
Trace: SMTL(AUC) vs. ECC	7.45E-02	5.06E-03	3.67E-02	1.66E-02
Micro F_1				
$\ell_{2,1}$: SMTL(F_1) vs. MTL-CLS	5.06E-03	5.93E-02	5.06E-03	9.34E-03
$\ell_{2,1}$: SMTL(F_1) vs. MTL-REG	5.06E-03	5.06E-03	5.06E-03	2.84E-02
Trace: SMTL(F_1) vs. MTL-CLS	5.06E-03	1.66E-02	5.06E-03	9.26E-02
Trace: SMTL(F_1) vs. MTL-REG	5.06E-03	5.06E-03	5.06E-03	6.91E-03
$\ell_{1,1}$: SMTL(F_1) vs. MTL-CLS	5.06E-03	4.69E-02	5.06E-03	5.93E-02
$\ell_{1,1}$: SMTL(F_1) vs. MTL-REG	5.06E-03	5.06E-03	5.06E-03	2.84E-02
Trace: SMTL(F_1) vs. RAkEL	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(F_1) vs. MLCSSP	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(F_1) vs. AdaBoostMH	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(F_1) vs. HOMER	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(F_1) vs. BR	5.06E-03	5.06E-03	9.34E-03	1.69E-01
Trace: SMTL(F_1) vs. LP	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL((F_1) vs. ECC	5.06E-03	5.06E-03	5.06E-03	2.03E-01
Macro F_1				
$\ell_{2,1}$: SMTL(F_1) vs. MTL-CLS	9.34E-03	5.06E-03	5.06E-03	4.69E-02
$\ell_{2,1}$: SMTL(F_1) vs. MTL-REG	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(F_1) vs. MTL-CLS	9.34E-03	5.06E-03	5.06E-03	5.93E-02
Trace: SMTL(F_1) vs. MTL-REG	5.06E-03	5.06E-03	5.06E-03	5.06E-03
$\ell_{1,1}$: SMTL(F_1) vs. MTL-CLS	9.34E-03	5.06E-03	5.06E-03	1.14E-01
$\ell_{1,1}$: SMTL(F_1) vs. MTL-REG	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(F_1) vs. RAkEL	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(F_1) vs. MLCSSP	5.06E-03	5.06E-03	5.06E-03	7.45E-02
Trace: SMTL(F_1) vs. AdaBoostMH	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(F_1) vs. HOMER	5.06E-03	3.67E-02	5.06E-03	5.06E-03
Trace: SMTL((F_1) vs. BR	5.06E-03	5.06E-03	5.06E-03	2.84E-02
Trace: SMTL((F_1) vs. LP	5.06E-03	5.06E-03	1.69E-01	8.79E-01
Trace: SMTL((F_1) vs. ECC	4.69E-02	5.06E-03	1.66E-02	9.59E-01

Fig. 2: Comparison results on Segmentation, Emotions and Optdigits w.r.t. Macro F1 (up) and Micro F1 (down).

metrics like F-score or AUC are more effective for evaluation than the misclassified errors. This is one of the reasons to motivate the proposed SMTL method in this paper. In MTL, the imbalance can be measured by firstly calculating the imbalance ratio in each individual task (i.e., $\frac{the\ number\ of\ positive\ instances}{the\ number\ of\ negative\ instances}$ for each task), and then averaging these ratios.

We conduct simulated experiments on 3 datasets (Segmentation, Emotions and Optdigits) to investigate the characteristics of the proposed SMTL methods on imbalanced data. In each dataset, we generate an imbalanced dataset by randomly selecting (with replacement) the positive and negative samples from the original dataset, with the ratio $1:1$, $1:5$ and $1:10$, respectively. As can be seen in Fig. 1 and Fig. 2, in most cases, the proposed SMTL variants consistently outperform the baseline method. For example, On Emotions with the

TABLE 8: Wilcoxon's test: p-values of SMTL against the baselines

Two methods for comparison	Cal500	Yeast	Emotions	Scene	Flags
Average AUC					
$\ell_{2,1}$: SMTL(AUC) vs. MTL-CLS	5.06E-03	5.06E-03	1.69E-01	**1.25E-02**	**1.25E-02**
$\ell_{2,1}$: SMTL(AUC) vs. MTL-REG	5.06E-03	5.06E-03	**1.25E-02**	5.06E-03	5.06E-03
Trace: SMTL(AUC) vs. MTL-CLS	6.91E-03	5.06E-03	1.14E-01	4.69E-02	5.08E-01
Trace: SMTL(AUC) vs. MTL-REG	5.06E-03	5.06E-03	**1.25E-02**	5.06E-03	3.67E-02
$\ell_{1,1}$: SMTL(AUC) vs. MTL-CLS	5.06E-03	5.06E-03	**1.25E-02**	2.84E-02	**1.25E-02**
$\ell_{1,1}$: SMTL(AUC) vs. MTL-REG	5.06E-03	5.06E-03	**1.25E-02**	5.06E-03	**1.25E-02**
Trace: SMTL(AUC) vs. RAkEL	5.06E-03	5.06E-03	**1.25E-02**	2.84E-02	5.06E-03
Trace: SMTL(AUC) vs. MLCSSP	5.06E-03	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(AUC) vs. AdaBoostMH	5.06E-03	5.06E-03	5.06E-03	5.06E-03	**1.25E-02**
Trace: SMTL(AUC) vs. HOMER	5.06E-03	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(AUC) vs. BR	5.06E-03	5.06E-03	5.06E-03	9.26E-02	5.06E-03
Trace: SMTL(AUC) vs. LP	5.06E-03	6.91E-03	5.06E-03	7.21E-01	9.34E-03
Trace: SMTL(AUC) vs. ECC	5.06E-03	5.06E-03	5.06E-03	1.69E-01	9.59E-01
Micro F_1					
$\ell_{2,1}$: SMTL(F_1) vs. MTL-CLS	5.06E-03	6.91E-03	5.06E-03	5.06E-03	5.06E-03
$\ell_{2,1}$: SMTL(F_1) vs. MTL-REG	9.34E-03	2.84E-02	2.84E-02	5.06E-03	**1.25E-02**
Trace: SMTL(F_1) vs. MTL-CLS	5.06E-03	6.91E-03	5.06E-03	5.06E-03	4.45E-01
Trace: SMTL(F_1) vs. MTL-REG	5.06E-03	2.84E-02	**1.25E-02**	5.06E-03	**1.25E-02**
$\ell_{1,1}$: SMTL(F_1) vs. MTL-CLS	5.06E-03	5.06E-03	5.06E-03	5.06E-03	3.67E-02
$\ell_{1,1}$: SMTL(F_1) vs. MTL-REG	5.06E-03	2.84E-02	3.67E-02	5.06E-03	2.84E-02
Trace: SMTL(F_1) vs. RAkEL	5.06E-03	9.34E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(F_1) vs. MLCSSP	5.06E-03	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(F_1) vs. AdaBoostMH	5.06E-03	5.06E-03	5.06E-03	5.93E-02	5.06E-03
Trace: SMTL(F_1) vs. HOMER	5.06E-03	2.03E-01	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL((F_1) vs. BR	5.06E-03	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL((F_1) vs. LP	5.06E-03	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL((F_1) vs. ECC	5.06E-03	3.86E-01	2.41E-01	4.69E-02	8.79E-01
Macro F_1					
$\ell_{2,1}$: SMTL(F_1) vs. MTL-CLS	5.06E-03	5.06E-03	6.91E-03	3.67E-02	3.67E-02
$\ell_{2,1}$: SMTL(F_1) vs. MTL-REG	5.06E-03	5.06E-03	5.06E-03	5.06E-03	6.91E-03
Trace: SMTL(F_1) vs. MTL-CLS	5.06E-03	5.06E-03	5.06E-03	1.66E-02	5.93E-02
Trace: SMTL(F_1) vs. MTL-REG	5.06E-03	5.06E-03	5.06E-03	5.06E-03	5.06E-03
$\ell_{1,1}$: SMTL(F_1) vs. MTL-CLS	5.06E-03	5.06E-03	5.06E-03	3.33E-01	3.67E-02
$\ell_{1,1}$: SMTL(F_1) vs. MTL-REG	5.06E-03	5.06E-03	5.06E-03	5.06E-03	2.84E-02
Trace: SMTL(F_1) vs. RAkEL	5.06E-03	**1.25E-02**	9.34E-03	9.26E-02	5.06E-03
Trace: SMTL(F_1) vs. MLCSSP	3.67E-02	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(F_1) vs. AdaBoostMH	5.06E-03	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL(F_1) vs. HOMER	5.06E-03	5.06E-03	5.06E-03	5.06E-03	5.06E-03
Trace: SMTL((F_1) vs. BR	5.06E-03	5.06E-03	5.06E-03	1.69E-01	5.06E-03
Trace: SMTL((F_1) vs. LP	5.06E-03	5.06E-03	5.06E-03	5.75E-01	6.91E-03
Trace: SMTL((F_1) vs. ECC	5.06E-03	5.06E-03	5.06E-03	7.99E-01	6.91E-03

ratio of $\frac{negative\ samples}{positive\ samples} = 10 : 1$, the proposed SMTL indicates a relative increase of 9.7% / 12.9% / 11.1% over the baseline w. r. t. AUC / Macro F1 / Micro F1, respectively. In addition, with the ratio of $\frac{negative\ samples}{positive\ samples}$ increasing, the improvement of SMTL over the baseline method also increases.

6.4 Training Time Comparison

To investigate the training speed of the proposed method, we provide the running time comparison results in Table 9. We can see that the training time of SMTL is (less than 30 times) slower than the baseline methods. It is worth noting that the training time cost is not a critical issue in practice, because the training process is usually off-line.

TABLE 9: Training Time Comparison

method	training time of Optdigits	training time of Emotions	training time of Segmentation
SMTL($\ell_{1,1}$+AUC)	105.200s	30.001s	1.888s
SMTL($\ell_{1,1}$+F_1)	510.900s	29.797s	2.964s
MTL-CLS($\ell_{1,1}$)	356.200s	24.674s	2.023s
MTL-REG($\ell_{1,1}$)	19.030s	7.427s	0.450s
StructSVM	17.762s	46.468s	5.015s
RAkEL	28.428s	4.117s	4.310s
AdaBoostMH	17.157s	1.024s	0.641s
MLCSSP	121.779s	1.563s	6.410s
HOMER	20.643s	1.354s	0.880s
BR	20.852s	1.859s	1.835s
LP	16.131s	22.561s	2.103s
ECC	17.852s	2.834s	1.891s

7 CONCLUSION

In this paper, we developed Structured-MTL, a MTL method of optimizing evaluation metrics. To solve the optimization problem of Structured MTL, we developed an optimization procedure based on ADMM scheme. This optimization procedure can be applied to solving a large family of MTL problems with structured outputs.

In the future work, we plan to investigate Structured-MTL on problems other than classification (e.g., MTL for ranking problems). We also plan to improve the efficiency of Structured-MTL on large-scale learning problems.

REFERENCES

[1] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817-1853, 2005.

[2] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243-272, 2008.

[3] J-B. Bi, T. Xiong, S-P. Yu, M. Dundar, and R. Rao. An improved multi-task learning approach with applications in medical diagnosis. In *Machine Learning and Knowledge Discovery in Databases*, pages 117-132, 2008.

[4] W. Bi, J. Kwok. Efficient Multi-label Classification with Many Labels. *Proceedings of the 30th International Conference on Machine Learning*. 405-413, 2013.

[5] J. Borwein and A. Lewis. Convex Analysis and Nonlinear Optimization. *Springer*, 2006.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1):1-122, 2011.

[7] J-F. Cai, E. J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *UCLA CAM Report*, 2008.

[8] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41-75, 1997.

[9] J-H. Chen, J. Liu, and J-P. Ye. Learning incoherent sparse and low-rank patterns from multiple tasks. In *International Conference on Knowledge Discovery and Data Mining*, pages 1179-1188, 2010.

[10] J-H. Chen, J-Y. Zhou, and J-P. Ye. Integrating low-rank and group-sparse structures for robust multi-task learning. In *International Conference on Knowledge Discovery and Data Mining*, pages 42-50, 2011.

[11] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615-637, 2005.

[12] R-E. Fan, K-W. Chang, C-J. Hsieh, X-R. Wang, C-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871-1874, 2008.

[13] P-H. Gong, J-P. Ye, and C-S. Zhang. Robust multi-task feature learning. In *International Conference on Knowledge Discovery and Data Mining*, pages 895-903, 2012.

[14] N. Gornitz, C. Widmer, G. Zeller, A. Kahles, S. Sonneburg, and G. Ratsch. Hierarchical Multitask Structured Output Learning for Large-Scale Sequence Segmentation. In *Advances in Neural Information Processing Systems*, 2011.

[15] X. Gu, F-L. Chung, H. Ishibuchi, and S-T. Wang. Multitask Coupled Logistic Regression and Its Fast Implementation for Large Multitask Datasets. In *IEEE Transactions on Cybernetics*, 45(9): 1953-1966, 2015.

[16] B. He, X. Yuan On the O(1/n) Convergence Rate of the Douglas-Rachford Alternating Direction Method. *SIAM Journal on Numerical Analysis*, 50(2): 700-709, 2012

[17] S-W. Ji and J-P. Ye. An Accelerated Gradient Method for Trace Norm Minimization. In *International Conference on Machine Learning*, pages 457-464, 2009

[18] Y-Z. Jiang, F-L. Chung, H. Ishibuchi, Z-H. Deng, and S-T. Wang. Multitask TSK Fuzzy System Modeling by Mining Intertask Common Hidden Structure. In *IEEE Transactions on Cybernetics*, 45(3): 548-561, 2015.

[19] T. Grigorios, S-X. Eleftherios, V. Jozef, and V. Ioannis. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414, 2011.

[20] T. Joachims. A Support Vector Method for Multivariate Performance Measures. In *International Conference on Machine Learning*, 2005.

[21] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *International Conference on Machine Learning*, pages 521-528, 2011.

[22] S. Kim and E. P. Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *International Conference on Machine Learning*, pages 543-550, 2010.

[23] H-J. Lai, Y. Pan, C. Liu, L. Lin, J. Wu Sparse Learning-to-rank via an Efficient Primal-Dual Algorithm. *IEEE Transactions on Computers*, 62(6):1221-1233, 2013

[24] H-J. Lai, Y. Pan, Y. Tang, R. Yu FSMRank: Feature Selection Method for Learning to Rank. *IEEE Transactions on Neaural Networks and Learning Systems*, 24(6):940-952, 2013

[25] Z-C. Lin, M-M. Chen, and Y. Ma. The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrix. Technical Report, UIUC, October 2009.

[26] A-A. Liu, Y-T. Su, P-P. Jia, Z. Gao, T. Hao, Z-X. Yang. Multiple/Single-View Human Action Recognition via Part-Induced Multitask Structural Learning. *IEEE transactions on cybernetics*, 45(6): 1194-1208, 2016.

[27] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*, pages 663-670, 2010.

[28] J. Liu, S-W. Ji, and J-P. Ye. Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization. In *Conference on Uncertainty in Artificial Intelligence*, pages 339-348, 2009.

[29] X-Q. Lu, X-L. Li, and L-C. Mou. Semi-Supervised Multitask Learning for Scene Recognition. In *IEEE Transactions on Cybernetics*, 45(9): 1967-1976, 2015.

[30] G. Obozinski, B. Taskar, and M.I. Jordan. Multi-task feature selection. Technical report, Statistics Department, UC Berkeley, 2006.

[31] Y. Pan, H-J. Lai, C. Liu, S-C. Yan. A Divide-and-Conquer Method for Scalable Low-Rank Latent Matrix Pursuit. In *International Conference on Computer Vision and Pattern Recognition*, 2013.

[32] Y. Pan, H-J. Lai, C. Liu, Y. Tang, S-C. Yan. Rank Aggregation via Low-Rank and Structured-Sparse Decomposition. In *AAAI Conference on Artificial Intelligence*, 2013.

[33] Y. Pan, R-K. Xai, J. Yin, N. Liu. A Divide-and-Conquer Method for Scalable Robust Multitask Learning. In *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3163-3175, 2015.

[34] N. Quadrianto, A. Smola, T. Caetano, S. Vishwanathan, and J. Petterson. Multitask learning without label correspondences. In *Advances in Neural Information Processing Systems*, pages 1957-1965, 2010.

[35] J. Read, B. Pfahringer, G. Holmes and E. Frank. Classifier Chains for Multi-label Classification. *Machine learning*, 85(3): 333-359, 2011.

[36] R.M. Rifkin and R.A. Lippert. Value Regularization and Fenchel Duality. *Journal of Machine Learning Research*, 8:441-479, 2007.

[37] R. E. Schapire, Y. Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*. 39(2):135-168, 2000

[38] S.S. Shwartz and Y. Singer. On the Equivalence of Weak Learnability and Linear Separability: New Relaxations and Efficient Boosting Algorithms *MachineLearning Journal*, vol. 80, no. 2, pp. 141-163, 2010.

[39] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support Vector Machine Learning for Interdependent and Structured Output Spaces. In *International Conference on Machine Learning*, 2004.

[40] G. Tsoumakas, I. Katakis and I. Vlahavas. Effective and Efficient Multilabel Classification in Domains with Large Number of Labels. *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data*. 30-44, 2008.

[41] G. Tsoumakas, I. Katakis and I. Vlahavas. Random k-Labelsets for Multi-Label Classification. *IEEE Transactions on Knowledge and Data Engineering*. 23(7):1079-1089, 2011.

[42] E. Gibaja, S. Ventura. A tutorial on multilabel learning. *ACM Computing Surveys*, 47(3): 52, 2015.

[43] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6): 80-83, 1945.

[44] R-K. Xia, Y. Pan, L. Du, J. Yin. Robust Multi-View Clustering via Low-Rank and Sparse Decomposition. In *AAAL Conference on Artificial Intelligence*, 2014.

[45] R-K. Xia, Y. Pan, H-J. Lai, C. Liu, S-C. Yan. Supervised Hashing for Image Retrieval via Image Representation Learning. In *AAAL Conference on Artificial Intelligence*, 2014.

[46] Y. Yang, Z-G. Ma, Y. Yang, F-P. Nie, and H-T. Shen. Multitask Spectral Clustering by Exploring Intertask Correlation. In *IEEE Transactions on Cybernetics*, 45(5): 1069-1080, 2015.

[47] Y-J. Yin, D. Xu, X-G. Wang, and M-R. Bai. Online State-Based Structured SVM Combined With Incremental PCA for Robust Visual Tracking. In *IEEE Transactions on Cybernetics*, 45(9): 1988-2000, 2015.

[48] J. Yu, D-C. Tao, M. Wang, and Y. Rui. Learning to Rank Using User Clicks and Visual Features for Image Retrieval. In *IEEE Transactions on Cybernetics*, 45(4): 767-779, 2015.

[49] K. Yu, V. Tresp, and A. Schwaighofer. Learning gaussian processes from multiple tasks. In *International Conference on Machine Learning*, pages 1012-1019, 2005.

[50] Y. Yue, T. Finley, F. Radlinski, T. Joachims. A Support Vector Method for Optimizing Average Precision. In *International Conference on Research and Development in Information Retrieval*, 2007.

[51] J. Zhang, Z. Ghahramani, and Y-M. Yang. Learning multiple related tasks using latent independent component analysis. In *Advances in Neural Information Processing Systems*, pages 1585-1592, 2006.

[52] W-Q. Zhao, Q-G Meng and P. W. H. Chung. A Heuristic Distributed Task Allocation Method for Multivehicle Multitask Problems and Its Application to Search and Rescue Scenario. *IEEE transactions on cybernetics*, 46(4): 902-915, 2016.

[53] J-Y. Zhou, J-H. Chen, and J-P. Ye. Clustered multi-task learning via alternating structure optimization. In *Advances in Neural Information Processing Systems*, pages 702-710, 2011.