ROM: A Robust Online Multi-Task Learning Approach

Chi Zhang*, Peilin Zhao§, Shuji Hao*, Yeng Chai Soh†, Bu Sung Lee ‡
*IGS, Nanyang Technological University, Singapore. Email: {czhang024,haos0001}@e.ntu.edu.sg
§Artificial Intelligence Department, Ant Financial, China. Email: peilin.zpl@antfin.com
†EEE, Nanyang Technological University, Singapore. Email: eycsoh@ntu.edu.sg
‡SCSE, Nanyang Technological University, Singapore. Email: ebslee@ntu.edu.sg

***Abstract*—A series of online multi-task learning (OMTL) algorithms have been proposed to avoid the expensive training cost and poor adaptability of traditional batch multi-task learning (MTL) algorithms in recent years. However, these OMTL algorithms usually assume that all tasks are closely related, which may not hold in practical scenarios. More importantly, their theoretical reliability is weakened due to the lack of proof on the cumulative regrets. To overcome these limitations, we present a robust online multi-task classification framework (ROM) and its two optimization algorithms (ROM-PGD, ROM-RDA). The proposed algorithms can not only automatically capture the common features among all tasks and individual features for each task, but also identify the potential existence of outlier task. Theoretically, we prove that the regret bounds of these two algorithms are sub-linear compared with the best separating algorithm in hindsight. Empirical studies on both synthetic and real-world datasets also demonstrate the effectiveness of our proposed algorithms when compared with the state-of-the-art OMTL algorithms.**

I. INTRODUCTION

Multi-task learning (MTL) aims at improving the generalization performance by learning multiple tasks in parallel and exploiting their intrinsic relationship. *Batch* MTL methods have been intensively studied in recent years [1], [2], and examples of its application can be found in handwritten digits recognition [3], disease prediction [4] and landmine detection [5]. However, MTL algorithms usually assume that all the training sets are provided in advance and construct models based on the whole dataset, leading to both expensive training cost and large storage demand. Besides, the poor adaptability of MTL algorithms in the changing environment makes it infeasible for real-time applications (e.g., stock market and business management). To overcome these limitations, researchers begin to design multi-task learning algorithms under the *online* setting. Online learning has long been proved to be an effective way in various literature [6]–[10], where data comes in a sequential order and models only deal with one instance on each round. A series of research [11]–[14] have verified its efficiency, efficacy and strong adaptability when applied to multi-task learning.

The key assumption for most OMTL algorithms is that all tasks are closely related and their hidden relationship is constrained through certain frameworks. Existing OMTL algorithms either restrict tasks' relationship by modeling a presumed interaction matrix or boost individual task's performance by leveraging a global model. However, it's not feasible to ensure that all tasks are closely related since the *relationship* among tasks is usually *unknown* before the learning process and *outlier tasks* often exist. Outlier task often fails to interact with other tasks or shifts the 'common center' towards its position, and therefore decreases OMTL algorithm's overall performance. Besides, most of the previous OMTL algorithms are only evaluated by empirical studies and few of them provide *theoretical analysis* to guarantee their cumulative regrets.

These limitations of existing algorithms strongly motivate our research on the robustness of online multi-task learning which would be guaranteed by theoretical proof. In this article, we propose a new framework (ROM) by considering three essential elements of OMTL: a **shared matrix** to construct a global learning model for all tasks, an **individual matrix** to store each task's specific features, and a **robust matrix** to discover outlier tasks. Closed-form solutions are provided based on the columns of these matrices on each round to avoid computation on matrix level, while non-smooth terms are optimized by two different algorithms, i.e, proximal gradient descent (PGD) and regularized dual averaging (RDA). Theoretical justifications for both algorithms are provided by comparing their cumulative regrets with the underlying best classification method, leading to a sub-linear regret in the form of $\mathcal{O}(\sqrt{T})$. With proper assumptions, these two regrets can identically achieve the same upper bound: $2G_*D\sqrt{T}$, where G_* and D are constants. The robustness of our proposed algorithms is fully examined by synthetic datasets, while experiments on real-world data also demonstrate the effectiveness of these two algorithms by comparing with the state-of-the-art OMTL algorithms.

The rest of this paper is organized as follows: Section II proposes our problem setting, together with two optimization solutions and their implementation details. Section III offers our theoretical justification for the two proposed algorithms. Section IV conducts a series of empirical studies to evaluate the performance of the proposed algorithms. Section V concludes this paper.

Notations: Lower-case letters ($\alpha, \beta, \cdots$) are used as scalars, and lower-case bold letters ($\mathbf{w}, \mathbf{q}, \cdots$) are used as vectors. Upper letters (U, P, Q) denote matrices, and their Euclidean and Frobenius norms are denoted by $\|\cdot\|$ and $\|\cdot\|_F$ respectively. Specially, (q,p)-norm of $A \in \mathrm{R}^{d\times m}$ represents

$\|A\|_{q,p} = (\sum_{i=1}^{m} \|\mathbf{a}_i\|_q^p)^{\frac{1}{p}}$. For multi-task learning, superscript denotes the number of task and subscript denote the number of round (e.g. $\mathbf{w}_t^i$ denotes the weight vector for i-th task on t-th round). Partial derivative of $\mathbf{g}_t$ with respect to $\mathbf{u}$ on round t is represented as $\mathbf{g}_{t,\mathbf{u}}$ (i.e., $\mathbf{g}_{t,\mathbf{u}} = \partial \mathbf{g}_t / \partial \mathbf{u}$).

II. ROBUST ONLINE MULTI-TASK CLASSIFICATION (ROM)

In this section, we first introduce the general setting for online multi-task classification problem. Based on the setting we propose our ROM framework, followed by two separate optimization solutions.

A. Problem Setting

Suppose we are given m tasks for binary classification in parallel, and the algorithm observes m instances $X_t = \left[\mathbf{x}_t^1\ \mathbf{x}_t^2\ \cdots\ \mathbf{x}_t^m\right] \in \mathbb{R}^{d\times m}$ on round t. For the i-th task, the algorithm makes a prediction based on the i-th column of its weight matrix $W_t \in \mathbb{R}^{d\times m}$, denoted as $\mathbf{w}_t^i$, $\hat{y}_t^i = \text{sign}(\hat{p}_t^i) = \text{sign}(\mathbf{w}_t^{i\top}\mathbf{x}_t^i)$. Then the true label y_t^i is revealed, and algorithm suffers a loss based on its prediction $\hat{p}_t^i$ and the true label y_t^i. In this article, we adopt the hinge loss for simplicity,

$$f_t^i(\mathbf{w}_t^i) = [1 - y_t^i \hat{p}_t^i]_+ = \max(0, 1 - y_t^i \mathbf{w}_t^{i\top}\mathbf{x}_t^i).$$

A regularization term $r(W_t)$ is then added to the empirical loss $\sum_{i=1}^{m}[f_t^i(\mathbf{w}_t^i)]$, aiming at introducing additional information to set restrictions for the smoothness or bound on the vector space norm. Therefore, the objective function for m tasks on round t is defined as,

$$\ell_t(W_t) = F_t(W_t) + r(W_t) = \sum_{i=1}^{m}[f_t^i(\mathbf{w}_t^i)] + r(W_t).$$

Our goal is minimizing the cumulative loss over all rounds T $\sum_{t=1}^{T}\ell_t(W_t)$ compared with the best classifier W_* in hindsight, where W_* is defined as $W_* = \text{argmin}_W \sum_{t=1}^{T}\ell_t(W)$. Their discrepancy is also known as **regret**,

$$\begin{aligned} R_T &= \sum_{t=1}^{T}\ell_t(W_t) - \sum_{t=1}^{T}\ell_t(W_*) \qquad (1)\\ &= \sum_{t=1}^{T}\sum_{i=1}^{m}[f_t^i(\mathbf{w}_t^i) - f_t^i(\mathbf{w}_*^i)] + \sum_{t=1}^{T}[r(W_t) - r(W_*)], \end{aligned}$$

where $\mathbf{w}_*^i$ denotes the i-th column of W_*.

B. Proposed Formulation

Following above settings, we propose a Robust Online Multi-task (ROM) Classification framework to alleviate the requirement of task relatedness and consider individual features for each task. Specially, the weight matrix of OMTL is formulated by considering three independent factors: a universal matrix $U \in \mathbb{R}^{d\times 1}$ to capture underlying shared information, an individual matrix $P \in \mathbb{R}^{d\times m}$ to store each task's specialized feature and an outlier matrix $Q \in \mathbb{R}^{d\times m}$ to tolerate the weak-related tasks. Therefore, the predictor for i-th task can be written as

$$\mathbf{w}^i = \mathbf{u} + \mathbf{p}^i + \mathbf{q}^i,$$

where $\mathbf{p}^i$ and $\mathbf{q}^i$ refer to the i-th column of the weight matrices P and Q.

On round t, we select the optimal matrix minimizing the current loss as weight matrix for the next round,

$$W_{t+1} = \underset{W}{\text{argmin}}\, F_t(W_t) + r(W_t), \qquad (2)$$

where the empirical loss $F_t(W_t)$ and regularization term $r(W_t)$ are calculated as

$$\begin{aligned} F_t(W_t) &= \sum_{i=1}^{m} f_t^i(\mathbf{w}_t^i) = \sum_{i=1}^{m}[1 - y_t^i(\mathbf{u}_t + \mathbf{p}_t^i + \mathbf{q}_t^i)^T\mathbf{x}_t^i]_+,\\ r(W_t) &= \Psi(\|U_t\|_F^2, \|P_t\|_F^2, \|Q_t\|_{2,1}). \end{aligned}$$

The intuition behind $r(W_t)$ is that: 1) we utilize the Frobenius norm to restrict the sizes of global model U_t and the individual feature matrix P_t; 2) a group Lasso penalty is applied on the columns of Q_t to generate a sparse outlier matrix. Specifically, L-$(2,1)$ norm selects features with joint sparsity and favors zero column vectors: if the column vector only contains zero components, the corresponding task will obey the universal model; if non-zero value occurs in a certain column of the matrix Q, the corresponding task can be identified as an outlier and its extra information will be stored in the outlier matrix. These three norms are concatenated through a function Ψ for the convenience of optimization.

Our objective function (2) is based on three independent weight matrices, and direct optimization on matrix slows down the computation speed [15]. Besides, it also contains a non-smooth term $\|Q\|_{2,1}$ and direct calculation with sub-gradient methods leads to slow convergence rate and a lack of an implementable stopping criterion [16]. Therefore, we decompose it into weight vectors and give closed-form solutions based on two independent optimization methods, i.e., proximal gradient descent (PGD) and regularized dual averaging (RDA).

C. Proximal Gradient Descent ROM (ROM-PGD)

We first introduce ROM algorithm based on proximal gradient descent (PGD) method. Proximal gradient method refers to a generalized form of projection used to solve non-differentiable convex optimization problems. The most natural choice for $r(W)$ in PGD optimization would be a linear combination of regularization terms

$$r_1(W) = \frac{\alpha}{2}\|U\|_F^2 + \frac{\beta}{2}\|P\|_F^2 + \gamma\|Q\|_{2,1},$$

for its simplicity and low cost to calculate the gradient.

Low computation cost when dealing with large scale data is one of the key advantages of online learning algorithms, while algorithms based on matrix computations usually scale to $\mathcal{O}(d^3)$. Instead of optimizing on the matrix level, we start by decomposing the weight matrices into weight vectors for each individual task, and then solve the non-smooth part by employing a projected gradient scheme [17], [18].

Lemma 1: By using first-order Taylor expansion of F_t, minimizer of Eq (2) can be derived by the following equations

$$\mathbf{u}_{t+1} = \underset{\mathbf{u}}{\text{argmin}} \sum_{i=1}^{m}\left\langle \mathbf{g}_{t,\mathbf{u}_t}^i, \mathbf{u} - \mathbf{u}_t\right\rangle + \frac{m}{2\eta}\|\mathbf{u} - \mathbf{u}_t\|_2^2 + \frac{\alpha}{2}\|\mathbf{u}\|_2^2, \qquad (3)$$

Algorithm 1 ROM-PGD Algorithm

Input: a sequence of instances $(\mathbf{x}_t^i, y_t^i), \forall t \in [1, T]$, and the parameters $\eta, \alpha, \beta, \gamma$
Initialize: $\mathbf{p}_0^i = 0, \mathbf{q}_0^i = 0$ for $\forall i \in [1, \cdots, m]$, $\mathbf{u}_0 = 0$;
for $t = 1, \cdots, T$ **do**
  for $i = 1, \cdots, m$ **do**
    Receive instance pair $(\mathbf{x}_t^i, y_t^i)$
    Calculate weight: $\mathbf{w}_t^i = \mathbf{u}_t + \mathbf{p}_t^i + \mathbf{q}_t^i$
    Predict $\hat{y}_t^i = \text{sign}(\mathbf{w}_t^i \cdot \mathbf{x}_t^i)$
    Compute the loss function $f_t^i(\mathbf{w}_t^i) = [1 - y_t^i \mathbf{w}_t^i \mathbf{x}_t^i]_+$
    if $f_t^i(\mathbf{w}_t^i) > 0$ **then**
      Update $\mathbf{p}_{t+1}^i$ according to Eq (7)
      Update $\mathbf{q}_{t+1}^i$ according to Eq (8) (10)
    else
      $\mathbf{p}_{t+1}^i = \mathbf{p}_t^i$ and $\mathbf{q}_{t+1}^i = \mathbf{q}_t^i$
    end if
  end for
  Update $\mathbf{u}_t$ according to Eq (6)
end for
Output: U, P, Q

$$\mathbf{p}_{t+1}^i = \underset{\mathbf{p}^i}{\text{argmin}} \left\langle \mathbf{g}_{t,\mathbf{p}_t^i}^i, \mathbf{p}^i - \mathbf{p}_t^i \right\rangle + \frac{1}{2\eta} \|\mathbf{p}^i - \mathbf{p}_t^i\|_2^2 + \frac{\beta}{2} \|\mathbf{p}^i\|_2^2, \tag{4}$$

$$\mathbf{q}_{t+1}^i = \underset{\mathbf{q}^i}{\text{argmin}} \left\langle \mathbf{g}_{t,\mathbf{q}_t^i}^i, \mathbf{q}^i - \mathbf{q}_t^i \right\rangle + \frac{1}{2\eta} \|\mathbf{q}^i - \mathbf{q}_t^i\|_2^2 + \gamma \|\mathbf{q}^i\|_2, \tag{5}$$

where $\mathbf{g}_{t,\mathbf{u}_t}^i = \partial f_t^i(\mathbf{w}_t^i)/\partial \mathbf{u}_t$, $\mathbf{g}_{t,\mathbf{p}_t^i}^i = \partial f_t^i(\mathbf{w}_t^i)/\partial \mathbf{p}_t^i$ and $\mathbf{g}_{t,\mathbf{q}_t^i} = \partial f_t^i(\mathbf{w}_t^i)/\partial \mathbf{q}_t^i$.

Eqs (3) and (4) are both convex and smooth, and their updating rules can be derived by setting the derivatives with respect to $\mathbf{u}_t$ and $\mathbf{p}_t^i$ to zero,

$$\mathbf{u}_{t+1} = \frac{1}{1 + \frac{\alpha\eta}{m}} \left[\mathbf{u}_t - \frac{\eta}{m} \sum_{i=1}^m \mathbf{g}_{t,\mathbf{u}_t}^i \right], \tag{6}$$

$$\mathbf{p}_{t+1}^i = \frac{1}{1 + \beta\eta} \left(\mathbf{p}_t^i - \eta \cdot \mathbf{g}_{t,\mathbf{p}_t^i}^i \right). \tag{7}$$

Eq (5) contains both differentiable and non-differentiable terms. Instead of using subgradient methods, we adopt the forward-backward splitting method [19] to solve a proximal operator problem in two steps:

$$\text{Step I} : \hat{\mathbf{q}}_t^i = \mathbf{q}_t^i - \eta \ \mathbf{g}_{t,\mathbf{q}_t^i}^i, \tag{8}$$

$$\text{Step II} : \mathbf{q}_{t+1}^i = \underset{\mathbf{q}^i}{\text{argmin}} \frac{1}{2\eta} \|\mathbf{q}^i - \hat{\mathbf{q}}_t^i\|_2^2 + \gamma \|\mathbf{q}^i\|_2, \tag{9}$$

where Step II admits a closed-form solution with the time complexity of $\mathcal{O}(d)$ [20],

$$\mathbf{q}_{t+1}^i = \max(0, 1 - \frac{\eta\gamma}{\|\hat{\mathbf{q}}_t^i\|_2}) \hat{\mathbf{q}}_t^i. \tag{10}$$

Therefore, $\mathbf{u}_t$, $\mathbf{p}_t^i$ and $\mathbf{q}_t^i$ are all updated through closed-form solutions. We summarize our ROM algorithm based on PGD in Algorithm 1.

D. Regularized Dual Averaging ROM (ROM-RDA)

In this part, we introduce ROM algorithm based on regularized dual averaging (RDA) method. RDA method refers to learning variables optimization by running average of all past subgradients [21], [22]. At step t, the algorithm receives a gradient or subgradient $\mathbf{g}_t$ and then calculates its average as

$$\overline{\mathbf{g}}_t = \sum_{\tau=1}^t \mathbf{g}_\tau = \frac{1}{t} \mathbf{g}_t + \frac{t-1}{t} \overline{\mathbf{g}}_{t-1}.$$

After the averaging process, a regularization term $r(\mathbf{w})$ is added to restrict the oscillation at each iteration. Therefore, the final updating rule is formulated as

$$\mathbf{w}_{t+1} = \underset{\mathbf{w}}{\text{argmin}} \left\{ \frac{1}{t} \sum_{\tau=1}^t \langle \mathbf{g}_\tau, \mathbf{w} \rangle + r(\mathbf{w}) \right\}. \tag{11}$$

To restrict the size of U and P and generate sparsity for Q, we use the following form of regularization terms,

$$r_2(W) = \frac{\alpha}{2} \|U\|_F^2 + \frac{\beta}{2} \|P\|_F^2 + \gamma \|Q\|_{2,1} + \frac{\theta_t}{2t} \left(\|U\|_F^2 + \|P\|_F^2 + \|Q\|_F^2 \right), \tag{12}$$

where $\{\theta_t\}_{t=1}^\infty$ is a nonnegative and nondecreasing input sequence.

Noting that the weight matrix W consists of three components U, P and Q, we formulate our proposed optimization ROM-RDA method in the following form,

$$(U_{t+1}, P_{t+1}, Q_{t+1}) \triangleq \underset{U,P,Q}{\text{argmin}} \frac{1}{t} \sum_{\tau=1}^t \left\langle \frac{\partial F_t}{\partial U_\tau}, U - U_\tau \right\rangle + \frac{1}{t} \sum_{\tau=1}^t \left\langle \frac{\partial F_t}{\partial P_\tau}, P - P_\tau \right\rangle + \frac{1}{t} \sum_{\tau=1}^t \left\langle \frac{\partial F_t}{\partial Q_\tau}, Q - Q_\tau \right\rangle + r_2(W). \tag{13}$$

Similar to Lemma 1, we decompose U_t, P_t and Q_t into their columns to avoid updating on matrix level and optimize these column vectors in closed-form solutions. The first step in RDA optimization is calculating the dual average of gradients or subgradients in the following forms,

$$\overline{\mathbf{g}}_{t,\mathbf{u}_t} = \frac{1}{t} \left(\sum_{i=1}^m \frac{\partial f_t^i}{\partial \mathbf{u}_t} \right) + \frac{t-1}{t} \overline{\mathbf{g}}_{t-1,\mathbf{u}_t}, \tag{14}$$

$$\overline{\mathbf{g}}_{t,\mathbf{p}_t^i} = \frac{1}{t} \left(\frac{\partial f_t^i}{\partial \mathbf{p}_t^i} \right) + \frac{t-1}{t} \overline{\mathbf{g}}_{t-1,\mathbf{p}_t^i}, \tag{15}$$

$$\overline{\mathbf{g}}_{t,\mathbf{q}_t^i} = \frac{1}{t} \left(\frac{\partial f_t^i}{\partial \mathbf{q}_t^i} \right) + \frac{t-1}{t} \overline{\mathbf{g}}_{t-1,\mathbf{q}_t^i}, \tag{16}$$

where $g_{t,*}$ represents the partial derivative of loss function f with respect to * on round t, and $\overline{g}_{t,*}$ is the dual average. Then the regularization term $r_2(W)$ is added, and with some simple algebra, solutions to Eq (13) can be substituted by

$$\mathbf{u}_{t+1} = \underset{\mathbf{u}}{\text{argmin}} \langle \overline{\mathbf{g}}_{t,\mathbf{u}_t}, \mathbf{u} \rangle + \frac{\alpha}{2} \|\mathbf{u}\|_2^2 + \frac{\theta_t}{2t} \|\mathbf{u}\|_2^2, \tag{17}$$

$$\mathbf{p}_{t+1}^i = \underset{\mathbf{p}^i}{\text{argmin}} \left\langle \overline{\mathbf{g}}_{t,\mathbf{p}_t^i}, \mathbf{p}^i \right\rangle + \frac{\beta}{2} \|\mathbf{p}^i\|_2^2 + \frac{\theta_t}{2t} \|\mathbf{p}^i\|_2^2, \tag{18}$$

$$\mathbf{q}_{t+1}^i = \underset{\mathbf{q}^i}{\text{argmin}} \left\langle \overline{\mathbf{g}}_{t,\mathbf{q}_t^i}, \mathbf{q}^i \right\rangle + \gamma \|\mathbf{q}^i\|_2 + \frac{\theta_t}{2t} \|\mathbf{q}^i\|_2^2. \tag{19}$$

Algorithm 2 ROM-RDA Algorithm

Input: a sequence of instances $(\mathbf{x}_t^i, y_t^i), \forall t \in [1, T]$, and the parameters $\alpha, \beta, \gamma, \theta_t$
Initialize: $\mathbf{p}_0^i = 0, \mathbf{q}_0^i = 0$ for $\forall i \in [1, m]$, $\mathbf{u}_0 = 0$;
Initialize: $\overline{\mathbf{g}}_{0,\mathbf{p}^i} = 0, \overline{\mathbf{g}}_{0,\mathbf{q}^i} = 0$ for $\forall i \in [1, m]$, $\overline{\mathbf{g}}_{0,\mathbf{u}} = 0$;
for $t = 1, \cdots, T$ **do**
  for $i = 1, \cdots, m$ **do**
    Receive instance pair $(\mathbf{x}_t^i, y_t^i)$
    Calculate weight: $\mathbf{w}_t^i = \mathbf{u}_t + \mathbf{p}_t^i + \mathbf{q}_t^i$
    Predict $\hat{y}_t^i = \text{sign}(\mathbf{w}_t^i \cdot \mathbf{x}_t^i)$
    Compute the loss function: $f_t^i(\mathbf{w}_t^i) = [1 - y_t^i \mathbf{w}_t^i \mathbf{x}_t^i]_+$
    Compute partial derivative average $\overline{\mathbf{g}}_{t,\mathbf{p}_t^i}$ as Eq (15)
    Update $\mathbf{p}_{t+1}^i$ according to Eq (21)
    Compute partial derivative average $\overline{\mathbf{g}}_{t,\mathbf{q}_t^i}$ as Eq (16)
    Update $\mathbf{q}_{t+1}^i$ according to Eq (22)
  end for
  Compute partial derivative average $\overline{\mathbf{g}}_{t,\mathbf{u}_t}$ as Eq (14)
  Update $\mathbf{u}_{t+1}$ according to Eq (20)
end for
Output: U, P, Q

The updating rules of Eqs (17) (18) can be easily derived as

$$\mathbf{u}_{t+1} = -\frac{1}{\alpha + \frac{\theta_t}{t}} \overline{\mathbf{g}}_{t,\mathbf{u}_t}, \tag{20}$$

$$\mathbf{p}_{t+1}^i = -\frac{1}{\beta + \frac{\theta_t}{t}} \overline{\mathbf{g}}_{t,\mathbf{p}_t^i}. \tag{21}$$

Denoting the optimal solution as $\mathbf{q}_{t+1}^i$, the problem of Eq (19) has a unique solution given by

$$\mathbf{q}_{t+1}^i = \max\left(0, 1 - \frac{\gamma}{\|\overline{g}_{t,\mathbf{q}_t^i}\|_2}\right)\left(-\frac{t \cdot \overline{g}_{t,\mathbf{q}_t^i}}{\theta_t}\right). \tag{22}$$

Once again, column vectors of three matrices U_t, P_t and Q_t are updated in closed-form solutions on round t. Algorithm 2 summarizes the generic algorithm ROM-RDA.

III. THEORETICAL ANALYSIS

We evaluate the performance of the proposed algorithms (ROM-PGD, ROM-RDA) by comparing their regrets with the best classification model in hindsight. Considering the regret R_T defined in Eq (1), we have the regrets of ROM-PGD and ROM-RDA in the following theorems respectively.

A. Regret of ROM-PGD

We first derive the upper bound of ROM-PGD's regret in the following Lemma.

Lemma 2: Assume that $\{U_t, P_t, Q_t\}$ are generated by Eqs (6) (7) and (10), the following inequality holds for any $W \in \Omega$:

$$R_T \leq \frac{1}{\eta}\|W - W_0\|_F^2 + r_1(W_0) + \frac{\eta}{2}\sum_{t=1}^{T}\|G_t(W_t)\|_F^2.$$

Theorem 3: Suppose $W_0 = 0$ (i.e., set $U_0, P_0, Q_0 = 0$) and the loss function f_t is Lipschitz continuous, there exists some constant G_* so that $\max_t \|G_t(W_t)\|_F \leq G_*$ is satisfied. Then we obtain

$$R_T \leq G_*\sqrt{2(\|U_*\|_F^2 + \|P_*\|_F^2 + \|Q_*\|_F^2)} \cdot \sqrt{T}$$

B. Regret of ROM-RDA

We first define a gap function δ_T and prove it to be an upper bound for our regret $R_T^{'}$.

Lemma 4: Denoting $\Delta_t = \sum_{i=1}^{m}\langle \mathbf{g}_t^i, \mathbf{w}_t^i - \mathbf{w}^i\rangle$ and defining the gap function δ_T as

$$\delta_T = \max_{W \in \Omega}\left\{\sum_{t=1}^{T}(\Delta_t + r_2(W_t)) - T \cdot r_2(W)\right\},$$

we have $R_T' \leq \delta_T$ holds.

Lemma 5: Following the same definition of G_* in theorem 3 and defining the feasible domain of W as $\mathcal{F}_D \triangleq \{W \in \Omega \mid \frac{1}{2}\|W\|_F^2 \leq D^2\}$, the upper bound of δ_T can be derived as:

$$\delta_T \leq \theta_T D^2 + \frac{G_*^2}{2}\sum_{\tau=1}^{T}\frac{1}{\theta_\tau}.$$

Theorem 6: By setting $\theta_\tau = \kappa\sqrt{\tau}$, ROM-RDA achieves a sub-linear regret $R_T' \leq \left(\kappa D^2 + \frac{G_*^2}{\kappa}\right)\sqrt{T}$.

C. Regret Comparison

Regret bound measures the performance of an online algorithm relative to the performance of a competing prediction mechanism, called a competing hypothesis [23]. By comparing the regret bounds of these two algorithms, we can see

1) R_T in theorem 3 and R_T' in theorem 6 both achieve a sub-linear regret $\mathcal{O}(\sqrt{T})$ when compared with the optimal classifier.

2) Recall the feasible domain for ROM-RDA is defined as $\{W \mid \frac{1}{2}\|W\|_F^2 \leq D^2\}$. By requiring the optimal solution in ROM-PGD as $\frac{1}{2}\|W^*\|_F^2 = \frac{1}{2}(\|U^*\|_F^2 + \|P^*\|_F^2 + \|Q^*\|_F^2) \leq D^2$, the regrets of these two algorithms achieve an identical upper bound $2G_*D\sqrt{T}$.

IV. EXPERIMENTS

In this section, we evaluate the performance of our proposed algorithms on one synthetic dataset and three different real-world datasets.

A. Baseline and Evaluation Metrics

In order to emphasize multi-task learning, we compare our proposed algorithms with a few variations of Perceptron [24] (abbreviated as PT) and Passive-Aggressive algorithms [6] (abbreviated as PA) in terms of global and individual learning.

- **Global learning** constructs *one* universal model shared by all tasks. Each task makes its prediction based on the global model and is authorized to modify the model.
- **Individual learning** employs m independent models. Each task updates its personalized model by its own instance and is denied access to other tasks' data.

We also compare our ROM-PGD and ROM-RDA with three state-of-the-art online multi-task algorithms in this part. We summarize these algorithms as follows,

1) **Multi-Perceptron**: online multi-task perceptrons with a fixed interaction matrix [13]. Interaction parameter b is tuned in $[0, \cdots, m]$ to achieve the best performance on each dataset.
2) **OMTL**: online multi-task classification based on an adaptive interaction matrix [14]. For each dataset we select the most effective one among these three algorithms provided in [14], denoted as OMTL-B, to represent the final performance .
3) **COML**: online multi-task learning through a collaborative structure [25]. COML constructs a global center for all tasks, and on each round the global model will be leveraged by each individual task to form its collaborative updating rule.

The performance of the algorithms is evaluated by their cumulative error rates, i.e., the ratio of mistakes over the total number of instances. Each experiment point consists of 10 runs, where the instances of each dataset are randomly shuffled. The final error rate reported is the average error rate of the 10 runs in an experiment. Parameters α, β, γ in ROM-PGD and ROM-RDA are tuned within a grid search $\{10^{-6}, \cdots, 10^{3}\}$ and learning rate η is tuned from 10^{-6} to 10^{-1}. θ_t in ROM-RDA is set to be $\kappa\sqrt{t}$ and κ is tuned within $\{10^{-3}, \cdots, 10^{3}\}$.

B. Synthetic Dataset

We first generate a synthetic dataset to show how these algorithms solve multi-task classification problems simultaneously and tackle the outlier task. Following [25], we construct 5 related tasks via a random walk with Gaussian increments. Specifically, the first 60 elements of $\mathbf{w}^1$ are set to be 1 and the last 40 components are set to be -1.5, i.e., $\mathbf{w}_1 = [1, \cdots, 1, -1.5, \cdots, -1.5]$. For $i = 2, 3, 4$, we set $\mathbf{w}^i = \mathbf{w}^{i-1} + \epsilon$ where $\epsilon \sim N(0, 0.09I)$. For $i = 5$, we set $\mathbf{w}^i = \mathbf{w}^{i-1} + \mu$ and $\mu \sim N(0, \delta^2 I)$, where δ^2 is a tuning parameter and will be $\delta^2 = 0.09, 1.00, 12.25$ in order to construct an outlier task.

Based on table I, we can observe that tuning δ^2 does not affect the performance of PT-individual and PA-individual, which can be treated as a *baseline*. When δ^2 is rather small (i.e., $\delta^2 = 0.09$), COML, ROM-PGD and ROM-RDA outperform the other algorithms on these five similar tasks, and significantly reduce the error rate from 14% to less than 8%. When further increasing δ^2, task 5 becomes an outlier task and ROM-PGD's performance is rather robust compared with its competitors. Such a phenomenon can be easily comprehended by the following explanation: task 1 to task 4 form a group learning system and their performance is improved through an OMTL framework while the last task receives limited information from the other tasks and is actually updated as an independent task. The performance of ROM-RDA is similar when $\delta^2 = 1.00$, but is less competitive under the extreme condition of $\delta^2 = 12.25$. These observations validate that our proposed algorithms not only benefit through online multi-task learning when the tasks are closely related, but also are *robust* on the noisy setting where outlier tasks exist.

Table I
CUMULATIVE ERROR RATE(%) ON SYNTHETIC DATASET

	Task 1	Task 2	Task 3	Task 4	Task 5 ($\delta^2 = 0.09$)
PT-Individual	14.10 ± 0.11	13.99 ± 0.17	13.95 ± 0.17	13.70 ± 0.13	13.88 ± 0.14
PA-Individual	13.79 ± 0.06	13.14 ± 0.07	12.40 ± 0.04	12.72 ± 0.03	12.21 ± 0.06
Multi-P	13.41 ± 0.21	12.99 ± 0.17	12.96 ± 0.19	13.10 ± 0.31	13.58 ± 0.22
OMTL-B	10.35 ± 0.34	10.06 ± 0.30	10.20 ± 0.21	11.51 ± 0.35	11.51 ± 0.22
COML	7.88 ± 0.12	7.41 ± 0.10	7.71 ± 0.17	7.43 ± 0.13	7.59 ± 0.15
ROM-PGD	7.89 ± 0.11	7.20 ± 0.13	7.72 ± 0.11	7.12 ± 0.12	6.75 ± 0.17
ROM-RDA	7.93 ± 0.14	7.81 ± 0.12	7.96 ± 0.10	7.35 ± 0.11	7.15 ± 0.13
	Task 1	Task 2	Task 3	Task 4	Task 5 ($\delta^2 = 1.00$)
PT-Individual	13.90 ± 0.14	14.03 ± 0.15	13.93 ± 0.14	13.76 ± 0.14	13.70 ± 0.17
PA-Individual	12.67 ± 0.03	12.28 ± 0.04	12.42 ± 0.06	12.89 ± 0.04	12.49 ± 0.07
Multi-P	13.95 ± 0.21	14.03 ± 0.27	13.93 ± 0.21	13.77 ± 0.30	13.71 ± 0.22
OMTL-B	12.05 ± 0.33	12.16 ± 0.34	12.01 ± 0.34	12.31 ± 0.21	13.96 ± 0.37
COML-1	11.18 ± 0.12	10.91 ± 0.12	10.96 ± 0.11	11.75 ± 0.14	12.89 ± 0.22
COML-2	8.40 ± 0.12	8.81 ± 0.14	8.35 ± 0.15	8.38 ± 0.15	20.48 ± 0.15
ROM-PGD	10.12 ± 0.11	9.72 ± 0.11	10.10 ± 0.14	10.40 ± 0.11	11.61 ± 0.14
ROM-RDA	10.36 ± 0.17	10.06 ± 0.21	10.21 ± 0.22	10.51 ± 0.18	11.52 ± 0.19
	Task 1	Task 2	Task 3	Task 4	Task 5 ($\delta^2 = 12.25$)
PT-Individual	14.24 ± 0.14	14.00 ± 0.13	14.19 ± 0.13	13.93 ± 0.14	13.80 ± 0.17
PA-Individual	12.41 ± 0.05	12.66 ± 0.07	12.27 ± 0.06	13.10 ± 0.07	12.99 ± 0.07
Multi-P	13.82 ± 0.24	14.15 ± 0.22	14.14 ± 0.22	14.03 ± 0.27	14.10 ± 0.34
OMTL-B	13.82 ± 0.30	13.85 ± 0.34	14.01 ± 0.24	13.93 ± 0.27	17.80 ± 0.51
COML-1	12.94 ± 0.12	12.15 ± 0.11	12.14 ± 0.14	12.89 ± 0.12	13.80 ± 0.11
COML-2	11.46 ± 0.22	10.62 ± 0.22	10.42 ± 0.20	11.55 ± 0.17	40.26 ± 0.51
ROM-PGD	10.64 ± 0.12	10.13 ± 0.12	10.16 ± 0.12	10.10 ± 0.14	13.61 ± 0.11
ROM-RDA	12.06 ± 0.27	12.16 ± 0.18	12.06 ± 0.26	12.30 ± 0.22	13.96 ± 0.19

C. Real World Datasets

We report on the performance of these algorithms on three commonly-used real-world datasets. (1) MHC-I: We adopt a subset of the original dataset[1] [26] with 12 human MHC-I alleles (A0201, A0202, A0203, A0206, A0301, A2402, A2902, A3002, A3101, A3301, A6801, A6802), where features are extracted through the bigram amino acid encoding. (2) TDT2. We construct an online multi-task classification problem by using one-against-all strategy on this dataset[2] [27]. (3) Each Movie[3]. We follow the method of [28] by randomly selecting 30 users who view exactly 200 movies and then choose 1783 users who viewed the same 200 movies and use their ratings as the features. Ratings are converted into binary classes, i.e., like or dislike.

Table II illustrates the cumulative error rates of different algorithms on real-world datasets. Apparently, all OMTL algorithms improve the overall performance compared with individual and global models. Among these algorithms, ROM-PGD and ROM-RDA continuously outperform their competitors and the performance stays rather stable, as indicated by the small deviations. Our theoretical analysis shows these two algorithms are upper bounded with the same regret, and it's *consistent* with our experiments, where the performance of these two algorithms is quite similar.

Fig 1 depicts the entire learning process for all OMTL algorithms of four typical MHC-I tasks. From the figure we can observe that the first two tasks go through insufficient learning, and their error rates keep decreasing until the end of the whole learning process. On the other hand, the following two tasks converge after a few iterations and their cumulative error rates are stable afterward. For these two different types of

[1] tools.iedb.org/main/datasets/

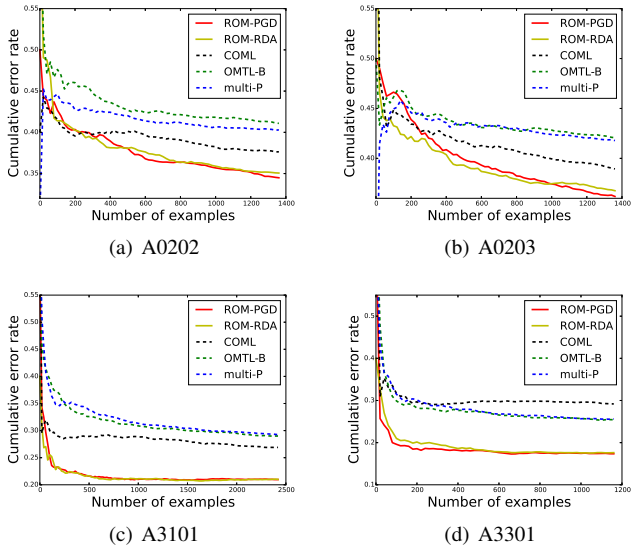
[2] http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html

[3] http://grouplens.org/datasets/eachmovie/

Table II
CUMULATIVE ERROR RATE(%) ON REAL-WORLD DATASETS

	MHC-I	TDT2	Each Movie
PT-individual	36.71 ± 0.23	6.65 ± 0.21	23.48 ± 0.57
PT-global	34.98 ± 0.20	39.22 ± 0.58	31.89 ± 0.16
PA-individual	36.62 ± 0.22	6.43 ± 0.10	23.40 ± 0.27
PA-global	33.99 ± 0.19	38.85 ± 0.47	31.13 ± 0.25
Multi-P	34.19 ± 0.19	6.20 ± 0.32	23.15 ± 0.48
OMTL-B	34.06 ± 0.19	5.99 ± 0.24	22.12 ± 0.61
COML	31.08 ± 0.22	3.80 ± 0.23	18.95 ± 0.52
ROM-PGD	$\mathbf{28.30 \pm 0.16}$	$\mathbf{3.34} \pm 0.07$	$\mathbf{18.49} \pm 0.19$
ROM-RDA	28.37 ± 0.15	3.41 ± 0.09	18.85 ± 0.34

learning, ROM-PGD and ROM-RDA consecutively beat other algorithms, and their cumulative error rates are roughly similar for all tasks, which is also consistent with the observations made in Fig II.



(a) A0202 (b) A0203

(c) A3101 (d) A3301

Figure 1. Cumulative error rate on MHC-I dataset along the entire learning process

V. ACKNOWLEDGE

This research is supported by the National Research Foundation, Prime Ministers Office, Singapore under its IDM Futures Funding Initiative. We also get support from "Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY) " and "Interdisciplinary Graduate School (IGS)".

VI. CONCLUSION

In this paper, we propose two robust online multi-task classification algorithms (ROM-PGD, ROM-RDA) to simultaneously capture the latent common features among multiple related tasks while detecting their individual characteristics and the potential existence of outlier tasks. The proposed algorithms enjoy closed-form updating solutions, which make them both efficient and effective.

REFERENCES

[1] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 109–117.

[2] C. Widmer, N. C. Toussaint, Y. Altun, and G. Rätsch, "Inferring latent task structure for multitask learning by multiple kernel learning," *BMC bioinformatics*, vol. 11, no. Suppl 8, p. S5, 2010.

[3] N. Quadrianto, J. Petterson, T. S. Caetano, A. J. Smola, and S. Vishwanathan, "Multitask learning without label correspondences," in *Advances in Neural Information Processing Systems*, 2010, pp. 1957–1965.

[4] J. Zhou, L. Yuan, J. Liu, and J. Ye, "A multi-task learning formulation for predicting disease progression," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 814–822.

[5] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram, "Multi-task learning for classification with dirichlet process priors," *The Journal of Machine Learning Research*, vol. 8, pp. 35–63, 2007.

[6] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *The Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

[7] N. Cesa-Bianchi, A. Conconi, and C. Gentile, "A second-order perceptron algorithm," *SIAM Journal on Computing*, vol. 34, no. 3, pp. 640–668, 2005.

[8] K. Crammer, M. Dredze, and F. Pereira, "Exact convex confidence-weighted learning," in *Advances in Neural Information Processing Systems*, 2009, pp. 345–352.

[9] P. Zhao, R. Jin, T. Yang, and S. C. Hoi, "Online auc maximization," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 233–240.

[10] S. C. Hoi, J. Wang, and P. Zhao, "Libol: A library for online learning algorithms," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 495–499, 2014.

[11] O. Dekel, P. M. Long, and Y. Singer, "Online multitask learning," in *Learning Theory*. Springer, 2006, pp. 453–467.

[12] G. Li, K. Chang, S. C. Hoi, W. Liu, and R. Jain, "Collaborative online learning of user generated content," in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 285–290.

[13] G. Cavallanti, N. Cesa-Bianchi, and C. Gentile, "Linear algorithms for online multitask classification," *The Journal of Machine Learning Research*, vol. 11, pp. 2901–2934, 2010.

[14] A. Saha, P. Rai, S. Venkatasubramanian, and H. Daume, "Online learning of multiple tasks and their relationships," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 643–651.

[15] C. D. Meyer, *Matrix analysis and applied linear algebra*. Siam, 2000, vol. 2.

[16] A. Bagirov, N. Karmitsa, and M. M. Mäkelä, *Introduction to Nonsmooth Optimization: theory, practice and software*. Springer, 2014.

[17] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical programming*, vol. 103, no. 1, pp. 127–152, 2005.

[18] 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*, vol. 3, no. 1, pp. 1–122, 2011.

[19] Y. Singer and J. C. Duchi, "Efficient learning using forward-backward splitting," in *Advances in Neural Information Processing Systems*, 2009, pp. 495–503.

[20] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient l 2, 1-norm minimization," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 2009, pp. 339–348.

[21] L. Xiao, "Dual averaging method for regularized stochastic learning and online optimization," in *Advances in Neural Information Processing Systems*, 2009, pp. 2116–2124.

[22] Y. Nesterov, "Primal-dual subgradient methods for convex problems," *Mathematical programming*, vol. 120, no. 1, pp. 221–259, 2009.

[23] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2011.

[24] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[25] G. Li, S. C. Hoi, K. Chang, W. Liu, and R. Jain, "Collaborative online multitask learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 8, pp. 1866–1876, 2014.

[26] B. Peters, H.-H. Bui, S. Frankild, M. Nielsen, C. Lundegaard, E. Kostem, D. Basch, K. Lamberth, M. Harndahl, W. Fleri *et al.*, "A community resource benchmarking predictions of peptide binding to mhc-i molecules," *PLoS Comput Biol*, vol. 2, no. 6, p. e65, 2006.

[27] D. Cai, X. Wang, and X. He, "Probabilistic dyadic data analysis with local and global consistency," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 105–112.