Learning Correlative and Personalized Structure for Online Multi-Task Classification

Peng Yang * Guangxia Li * Peilin Zhao * Xiaoli Li * Sujatha Das Gollapalli *

Abstract

Multi-Task Learning (MTL) can enhance the classifier's generalization performance by learning multiple related tasks simultaneously. Conventional MTL works under the offline or batch learning setting and suffers from the expensive training cost together with the poor scalability. To address such inefficiency issues, online learning technique has been applied to solve MTL problems. However, most existing algorithms for online MTL constrain task relatedness into a presumed structure via a single weight matrix, a strict restriction that does not always hold in practice. In this paper, we propose a general online MTL framework that overcomes this restriction by decomposing the weight matrix into two components: the first component captures the correlative structure among tasks in a low-rank subspace, and the second component identifies the personalized patterns for the outlier tasks. A projected gradient scheme is devised to learn such components adaptively. Theoretical analysis shows that the proposed algorithm can achieve a sub-linear regret with respect to the best linear model in hindsight. Experimental investigation on a number of real-world datasets also verifies the efficacy of our approach.

1 Introduction

Multi-Task Learning (MTL) aims to enhance the overall generalization performance by learning multiple related tasks simultaneously. It has been extensively studied from various points of view [1, 2, 3, 4]. As an example, the common tastes of users (tasks) with respect to movies (instances) can be harnessed into a movie recommender system using MTL [5]. Most MTL methods work under the offline learning setting where the training data for each task is available beforehand. However, offline learning methods are generally inefficient, as suggested by their high training cost and poor scalability. This is especially true when it comes to large-scale data and streaming data. As a remedy, MTL has been studied under the online learning setting, where the model works on a sequence of data by processing them one by one [6]. After updating model in each round, the current input data will be discarded to save space. As a result, online learning algorithms are efficient and scalable. They have been successfully applied on a couple of MTL applications [7, 8, 9, 10, 11].

In this paper, we investigate MTL under the online learning setting. Existing online MTL methods work under an assumption that all tasks are related with each other and simply constrain the relationship among multiple tasks via a presumed structure [7, 9]. However, such constraint may be too restrictive and rarely holds in the real-life applications, as the personalized tasks with individual traits often exist [12]. We attempt to address this drawback through a creative formulation of online MTL which incorporates two components. The first component captures a low-rank *correlative* structure over the related tasks. And the second one represents the *personalized* patterns specific to individual tasks.

Specifically, our algorithm learns a weight matrix which can be decomposed into two components as aforementioned. A trace-norm regularization is imposed on the first component to induce a low-rank *correlative* structure over the related tasks. A group lasso penalty over all individual tasks is applied on the second component through a regularization term to identify personalized patterns. The resulting non-smooth convex optimization problem is solved using an online projected gradient scheme. We show that a closed-form solution can be obtained for both the *correlative* and *personalized* components. This gives our algorithm two advantages: 1) it is very efficient as it can make prediction and update model in a real time manner; 2) it can achieve a good trade-off for learning several tasks jointly. We provide a theoretical evaluation of our algorithm by giving a proof that our algorithm can achieve a sub-linear regret compared with the best linear model in hindsight. We

*Institute for Infocomm Research, A*STAR, Singapore. {yangp,lig,zhaop,xlli,gollapallis}@i2r.a-star.edu.sg

also perform comparative experiment against a variety of state-of-the-art techniques on three real-life datasets. The experimental results confirm the efficacy of our method. To the best of our knowledge, we are the first to construct robust online MTL classifier by combining correlative parameters with personalized one.

The rest of this paper is organized as follows. Section 2 introduces related work. The problem setting and the proposed algorithm with analysis are presented in Section 3 and Section 4, respectively. Section 5 provides experimental results. Section 6 concludes this paper.

2 Related Work

We briefly introduce work related to MTL in the offline and online learning settings, respectively.

Conventional offline or batch learning MTL algorithms can be generally grouped into two categories: *explicit* parameter sharing and *implicit* parameter sharing. All tasks can be made to share some common parameters explicitly. Such common parameters include hidden units in neural networks [13], prior in hierarchical Bayesian models [14, 15], feature mapping matrix [16] and classification weight [17]. The shared structure can also be captured in an implicit way by imposing a low rank subspace [18, 19], or a common set of features [20, 21, 22].

Compared to offline learning, online learning technique is more efficient and suitable to handle massive and sequential data [23, 24, 25]. Task structure has been exploited by using a global loss function to evaluate the prediction [26], or assuming that a few experts can perform well on the entire task set [27, 28]. Instead of fixing task relationship via a presumed structure [9], a recent work introduces an adaptive interaction matrix which quantifies the task relevance [7]. A selective sampling strategy has also been applied to learn the online multitask model by querying a few informative labels [29]. The algorithm presented in this paper differs from existing ones in that it learns a common low-rank structure and individual outlier tasks simultaneously.

3 Problem Setting

In this section, we first describe our notations, followed by the problem setting of the online MTL.

3.1 Notations Lowercase letter is used as scalar, lowercase bold letter as vector, uppercase letter as element of a matrix, and bold-face uppercase letter as matrix. $\mathbf{x}_i$ and x_{ij} denote the i-th column and the (i,j)-th element of a matrix $\mathbf{X}$. Euclidean and Frobenius norms are denoted by $\|\cdot\|$ and $\|\cdot\|_F$. In particular, for every $q, p \geq 1$, we define the (q,p)-norm of $\mathbf{A} \in \mathbb{R}^{d\times m}$ as $\|\mathbf{A}\|_{q,p} = (\sum_{i=1}^{m} \|\mathbf{a}_i\|_q^p)^{\frac{1}{p}}$. The subdifferential set of a function f evaluated at $\mathbf{w}$ is denoted by $\partial f(\mathbf{w})$ and a particular subgradient by $f^{'}(\mathbf{w}) \in \partial f(\mathbf{w})$. When the function is differentiable, we denote its gradient by $\nabla f(\mathbf{w})$.

3.2 Problem Setting According to the online MTL setting, we are faced with m different but related classification problems, also known as tasks. Each task has a sequential instance-label pairs, i.e., $\{(\mathbf{x}_t^i, y_t^i)\}_{1\leq t\leq T}^{1\leq i\leq m}$, where $\mathbf{x}_t^i \in \mathbb{R}^d$ is a feature vector drawn from a single feature space shared by all tasks, and $y_t^i \in \{\pm 1\}$. The algorithm maintains m separate models in parallel, one for each of the m tasks. On round t, m instances $\{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^m\}$ are presented at one time. Given the i-th task instance $\mathbf{x}_t^i$, the algorithm makes prediction using a linear model $\mathbf{z}_t^i$, i.e., $\hat{y}_t^i = \text{sign}(\hat{p}_t^i)$, where $\hat{p}_t^i = \mathbf{z}_t^{i\top}\mathbf{x}_t^i$ and $\mathbf{z}_t^i$ is the weight parameter on round t. The true label y_t^i is revealed until then. A hinge-loss function is applied to evaluate the prediction,

$$f_t^i(\mathbf{z}_t^i) = [1 - y_t^i \hat{p}_t^i]_+ = [1 - y_t^i \mathbf{z}_t^i \cdot \mathbf{x}_t^i]_+, \tag{3.1}$$

where $[a]_+ = \max\{0, a\}$. The cumulative loss over all m tasks on round t is defined as

$$F_t(Z_t) = \sum_{i=1}^{m} f_t^i(\mathbf{z}_t^i), \tag{3.2}$$

where $Z_t = [\mathbf{z}_t^1, \ldots, \mathbf{z}_t^m] \in \mathbb{R}^{d\times m}$ is the weight matrix for all tasks. Inspired by the Regularized Loss Minimization (RLM) in which one minimizes an empirical loss plus a regularization term jointly [30], we formulate our online MTL to minimize the regret bound compared to the best linear model in hindsight,

$$R_\phi \triangleq \sum_{t=1}^{T} [F_t(Z_t) + g(Z_t)] - \inf_{Z\in\Omega} \sum_{t=1}^{T} [F_t(Z) + g(Z)], \tag{3.3}$$

where $\Omega \subset \mathbb{R}^d$ is a closed convex subset and the regularizer $g : \Omega \to \mathbb{R}$ is a convex regularization function that constraints Ω into simple sets, e.g., hyperplanes, balls, bound constraints, etc. For instance, $g(Z) = \|Z\|_1$ constrains Z into a sparse matrix. $\phi_t(Z) = F_t(Z) + g(Z)$ is a non-smooth convex function provided in the following.

4 Algorithm

We propose to solve the online MTL problem in (3.3) by two steps: 1) to learn the correlative and person-

alized patterns over multiple tasks; 2) to achieve an optimal solution for the proposed objective function.

4.1 Correlative and Personalized Structures As mentioned above, restricting task relatedness to a presumed structure via a single weight matrix [7] is too strict and not always plausible in practical applications. To overcome this problem, we decompose the weight matrix Z into two components: *correlative* matrix U and *personalized* matrix V, and define a new weight matrix,

$$\Omega = \{W | W = \begin{bmatrix} U \\ V \end{bmatrix}, U \in \mathbb{R}^{d\times m}, V \in \mathbb{R}^{d\times m}\}, \tag{4.4}$$

where $\mathbf{w}^i = \begin{bmatrix} \mathbf{u}^i \\ \mathbf{v}^i \end{bmatrix} \in \mathbb{R}^{2d}$ is the i-th column of the weight matrix $W = [\mathbf{w}^1, \ldots, \mathbf{w}^m] \in \mathbb{R}^{2d\times m}$. Denoted by matrix Z the summation of U and V, we obtain

$$Z = U + V = \left[I_d, I_d\right] W, \tag{4.5}$$

where $I_d \in \mathbb{R}^{d\times d}$ is an identity matrix. Given an instance $(\mathbf{x}_t^i, y_t^i)$, the algorithm makes prediction based on both the correlative and personalized parameters,

$$\begin{aligned} \hat{p}_t^i &= \mathbf{z}_t^i \cdot \mathbf{x}_t^i \overset{(4.5)}{=} (\left[I_d, I_d\right] \mathbf{w}_t^i) \cdot \mathbf{x}_t^i \\ &= (\mathbf{u}_t^i + \mathbf{v}_t^i)^\top \mathbf{x}_t^i, \end{aligned} \tag{4.6}$$

with the corresponding loss function,

$$f_t^i(\mathbf{z}_t^i) = f_t^i(\left[I_d, I_d\right] \mathbf{w}_t^i) = [1 - y_t^i(\mathbf{u}_t^i + \mathbf{v}_t^i)^\top \mathbf{x}_t^i]_+.$$

We thus can reformat the cumulative loss function over all m tasks with respect to W as

$$L_t(W_t) = F_t(Z_t) \overset{(4.5)}{=} F_t(\left[I_d, I_d\right] W_t). \tag{4.7}$$

To exploit the correlative and personalized patterns over multiple tasks, we impose a regularizer on U and V,

$$r(W) = g(\left[I_d, I_d\right] W) \triangleq \lambda_1 \|U\|_* + \lambda_2 \|V\|_{2,1}, \tag{4.8}$$

where λ_1 and λ_2 are non-negative trade-off parameters. The $\|U\|_*$ imposes a trace norm [18] on U to represent multiple tasks $(\mathbf{u}_i, i \in [1, m])$ by a small number (i.e. n) of the basis ($n \ll m$). Intuitively, a model performing well on one task is likely to perform well on the similar tasks. Thus, we expect a best model can be shared across several relative tasks. However, the assumption that all tasks are correlated may not hold in real applications. Thus, we impose the $(2, 1)$-norm [31] on V, which favors a few non-zero columns in the matrix V to capture the personalized tasks. Note that our algorithm is able to detect personalized patterns, unlike the algorithms [27, 28, 32]. Although prior work [12] considers detecting the personalized task, it is designed for offline setting, which is different from our algorithm since we learns personalized pattern adaptively with online technique.

Substitute equations (4.7) and (4.8) into the regret (3.3), the online MTL problem can be formatted as

$$R_\phi \triangleq \sum_{t=1}^{T} [L_t(W_t) + r(W_t)] - \inf_{W\in\Omega} \sum_{t=1}^{T} [L_t(W) + r(W)], \tag{4.9}$$

where $\phi_t(W) = L_t(W) + r(W)$ is a non-smooth convex function. We next show how to achieve an optimal solution to problem (4.9).

4.2 Online Task Relationship Learning Inspired by [33], we can solve the problem (4.9) by a subgradient projection,

$$\operatorname*{argmin}_{W\in\Omega} \|W - (W_t - \eta \nabla \phi_t(W_t))\|, \tag{4.10}$$

where $\eta > 0$ is the learning rate. In the following lemma, we show that the problem (4.10) can be turned into a linearized version of the proximal algorithm [34]. To do so, we first introduce a Bregman-like distance function [35],

$$B_\psi(W, W_t) = \psi(W) - \psi(W_t) - \langle W - W_t, \nabla\psi(W_t)\rangle,$$

where ψ is a differentiable and convex function.

LEMMA 4.1. *Assume $\psi(\cdot) = \frac{1}{2}\|\cdot\|_F^2$, then the algorithm (4.10) is equivalent to a linearized form with a step-size parameter $\eta > 0$,*

$$W_{t+1} = \operatorname*{argmin}_{W\in\Omega} \langle \nabla\phi_t(W_t), W - W_t\rangle + \frac{1}{\eta} B_\psi(W, W_t).$$

Intuitively, above linearized formulation prompts the model to perform well on the current instances as far as possible, while still staying close to the previous estimate. Instead of balancing this trade-off individually for each of the multiple tasks, we balance for all the tasks jointly. However, the subgradient of a composite function, i.e., $\nabla\phi_t(W_t) = \nabla L_t(W_t) + \nabla r(W_t)$ cannot lead to a desirable effect, since we should constrain the projected gradient (i.e. $W_t - \eta\nabla\phi_t(W_t)$) into a restricted set. To address this issue, we refine the optimization function by adding a regularizer on

W,

$$W_{t+1} \triangleq \operatorname*{argmin}_{W\in\Omega} \langle \nabla L_t(W_t), W - W_t\rangle + \frac{1}{\eta} B_\psi(W, W_t) + r(W). \tag{4.11}$$

Note that the formulation (4.11) is different from the Mirror Descent (MD) algorithm [36], since we do not *linearize* the regularizer r.

Given that $W = \left[U^\top, V^\top\right]^\top$, we show that the problem (4.11) can be reformatted in terms of U and V by the lemma below.

LEMMA 4.2. *Assume that* $\psi(W) = \frac{1}{2}\|W\|_F^2$ *and* $W = \begin{bmatrix} U \\ V \end{bmatrix}$, *the problem (4.11) turns into an equivalent form in terms of* U *and* V,

$$\begin{aligned}(U_{t+1}, V_{t+1}) \triangleq \operatorname*{argmin}_{U,V\in\Omega}\ & \lambda_1\|U\|_* + \lambda_2\|V\|_{2,1} \\ & + \frac{1}{2\eta_1}\|U - U_t\|_F^2 + \frac{1}{2\eta_2}\|V - V_t\|_F^2 \\ & + \langle \nabla_U L_t(U_t), U - U_t\rangle + \langle \nabla_V L_t(V_t), V - V_t\rangle,\end{aligned} \tag{4.12}$$

where the parameters η_1 *and* η_2 *control previous learned knowledge retained by* U *and* V.

Proof. Assume that $\psi(W) = \frac{1}{2}\|W\|_F^2$, we obtain:

$$\begin{aligned} B_\psi(W, W_t) &= \frac{1}{2}\|W\|_F^2 - \frac{1}{2}\|W_t\|_F^2 - \langle W - W_t, W_t\rangle \\ &= \frac{1}{2}\|W - W_t\|_F^2 \\ &\overset{(4.4)}{=} \frac{1}{2}\|U - U_t\|_F^2 + \frac{1}{2}\|V - V_t\|_F^2.\end{aligned} \tag{4.13}$$

The linearized gradient form can be rewritten as:

$$\begin{aligned} & \langle \nabla L_t(W_t), W - W_t\rangle \\ \overset{(4.4)}{=} & \left\langle \begin{bmatrix} \nabla_U L_t(U_t) \\ \nabla_V L_t(V_t) \end{bmatrix}, \begin{bmatrix} U - U_t \\ V - V_t \end{bmatrix} \right\rangle \\ = & \langle \nabla_U L_t(U_t), U - U_t\rangle + \langle \nabla_V L_t(V_t), V - V_t\rangle. \end{aligned} \tag{4.14}$$

Substitute (4.13), (4.14) and (4.8) into problem (4.11), we complete this proof.

We next present how to optimize above non-smooth convex problem with a closed-form solution.

4.3 Optimization Our objective function (4.12) is composite with both smooth and non-smooth terms. Although such composite problem can be solved by [37], composite function with linear constrains have not been investigated to solve the MTL problem. We employ a projected gradient scheme [38, 39] to optimize the problem (4.12). Specifically, by omitting the terms unrelated to U and V, the problem (4.12) can be rewritten as a projected gradient schema,

$$\begin{aligned}(U_{t+1}, V_{t+1}) = \operatorname*{argmin}_{U,V\in\Omega}\ & \frac{1}{2\eta_1}\|U - \hat{U}_t\|_F^2 + \lambda_1\|U\|_* \\ & + \frac{1}{2\eta_2}\|V - \hat{V}_t\|_F^2 + \lambda_2\|V\|_{2,1}.\end{aligned}$$

where

$$\hat{U}_t = U_t - \eta_1\nabla_U L_t(U_t), \quad \hat{V}_t = V_t - \eta_2\nabla_V L_t(V_t).$$

Due to the decomposability of above objective function, the solution for U and V can be optimized separately,

$$U_{t+1} = \operatorname*{argmin}_{U\in\Omega} \frac{1}{2\eta_1}\|U - \hat{U}_t\|_F^2 + \lambda_1\|U\|_*. \tag{4.15}$$

$$V_{t+1} = \operatorname*{argmin}_{V\in\Omega} \frac{1}{2\eta_2}\|V - \hat{V}_t\|_F^2 + \lambda_2\|V\|_{2,1}. \tag{4.16}$$

This has two advantages: 1) there is a close form solution for each update; 2) the update for the U and V can be performed in parallel.

4.3.1 Computation of U Inspired by [38], we show that the optimal solution to (4.15) can be obtained via solving a simple convex optimization problem in the following theorem.

THEOREM 1. *Denote by the eigendecomposition of* $\hat{U}_t = P\hat{\Sigma}Q^\top \in \mathbb{R}^{d\times m}$ *where* $r = rank(\hat{U}_t)$, $P \in \mathbb{R}^{d\times r}$, $Q \in \mathbb{R}^{m\times r}$, *and* $\hat{\Sigma} = diag(\hat{\sigma}_1, \ldots, \hat{\sigma}_r) \in \mathbb{R}^{r\times r}$. *Let* $\{\sigma_i\}_{i=1}^r, \sigma_i \geq 0$ *be the solution of the following problem,*

$$\min_{\{\sigma_i\}_{i=1}^r} \frac{1}{2\eta_1}\sum_{i=1}^r (\sigma_i - \hat{\sigma}_i)^2 + \lambda_1\sum_{i=1}^r \sigma_i. \tag{4.17}$$

Denote by $\Sigma = diag(\sigma_1, \ldots, \sigma_r) \in \mathbb{R}^{r\times r}$, *then the optimal solution to Eq. (4.15) is given by,*

$$U^* = P\Sigma Q^\top, \tag{4.18}$$

where the solution to Eq. (4.17) is $\sigma_i^* = [\hat{\sigma}_i - \eta_1\lambda_1]_+$ *for* $i \in [1, r]$.

4.3.2 Computation of V We rewrite (4.16) by solving an optimization problem for each column,

$$\min_{\mathbf{v}_i\in\mathbb{R}^d} \sum_{i=1}^m \frac{1}{2\eta_2}\|\mathbf{v}_i - \hat{\mathbf{v}}_i\|^2 + \lambda_2\sum_{i=1}^m \|\mathbf{v}_i\|_2. \tag{4.19}$$

Algorithm 1 ROMCO

Input: a sequence of instances $(\mathbf{x}_t^i, y_t^i,), \forall t \in [1, T]$, and the parameter η_1, η_2, λ_1 and λ_2.
Initialize: $\mathbf{u}_0^i = \mathbf{0}, \mathbf{v}_0^i = \mathbf{0}$ for $\forall i \in [1, m]$;
for $t = 1, \ldots, T$ **do**
  for $i = 1, \ldots, m$ **do**
    Receive instance pair $(\mathbf{x}_t^i\ y_t^i)$;
    Predict $\hat{y}_t^i = \text{sign}[(\mathbf{u}_t^i + \mathbf{v}_t^i) \cdot \mathbf{x}_t^i]$;
    Compute the loss function $f_t^i(\mathbf{w}_t^i)$;
  end for
  if $\exists i \in [1, m], f_t^i(\mathbf{w}_t^i) > 0$ **then**
    Update U_{t+1} with Eq. (4.18);
    Update V_{t+1} with Eq. (4.20);
  else
    $U_{t+1} = U_t$ and $V_{t+1} = V_t$;
  end if
end for
Output: $\mathbf{w}_T^i$ for $i \in [1, m]$

where $\hat{\mathbf{v}}_i \in \mathbb{R}^d$ denotes the i-th column of $\hat{V}_t = V_t - \eta_2 \nabla_V L_t(V_t) = [\hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_m]$. The above optimal operator problem (4.19) admits a close form solution with time complexity of $O(dm)$ [40],

$$\mathbf{v}_{t+1}^i = \max(0, 1 - \frac{\eta_2 \lambda_2}{\|\hat{\mathbf{v}}_t^i\|_2}) \hat{\mathbf{v}}_t^i, \quad \forall i \in [1, m]. \tag{4.20}$$

Thus, the two quantities V_t and U_t can be updated according to a closed-form solution on each round t.

4.4 Theoretical Analysis We call our algorithm "Robust Online Multi-task learning with Correlative and persOnalized structure", ROMCO for short. It is summarized in Alg. 1. Note that our algorithm adopts a mistake-driven update rule: it performs an update only when an error occurs ($\hat{y}_t^i \neq y_t^i$).

We next evaluate the performance of our algorithm ROMCO in terms of regret bound. We first show the regret bound of the algorithm (4.11) and its equivalent form in the following lemma, which is essentially the same as Theorem 2 in the paper [41]:

LEMMA 4.3. *Let $\{W_t\}$ be updated according to (4.11). Assume that $B_\psi(\cdot, \cdot)$ is α-strongly convex w.r.t. a norm $\|\cdot\|_p$ and its convex conjugate $\|\cdot\|_q$ with $\frac{1}{p} + \frac{1}{q} = 1$, for any $W^* \in \Omega$,*

$$R_\phi \leq \frac{1}{\eta} B_\psi(W^*, W_1) + r(W_1) + \frac{\eta}{2\alpha} \sum_{t=1}^{T} \|\nabla L_t(W_t)\|_q^2. \tag{4.21}$$

REMARK 4.1. *We show that the regret in (4.21) is $O(\sqrt{T})$ with respect to the best linear model in hindsight. Suppose that the functions F_t are Lipschitz continuous, then $\exists G_q$ such that $\max_t \|\nabla L_t(W_t)\|_q \leq G_q$. Then we obtain:*

$$R_\phi \leq \frac{1}{\eta} B_\psi(W^*, W_1) + r(W_1) + \frac{T\eta}{2\alpha} G_q^2.$$

We also assume that $r(W_1) = 0$. Then by setting $\eta = \sqrt{2\alpha B_\psi(W^*, W_1)}/(\sqrt{T} G_q)$, we have $R_\phi \leq \sqrt{2TB_\psi(W^*, W_1)} G_q/\sqrt{\alpha}$. Given that G_q is constant, setting $\eta \propto 1/\sqrt{T}$, we have $R_\phi = O(\sqrt{T})$.

LEMMA 4.4. *The general optimization problem (4.11) is equivalent to the two step process of setting*

$$\tilde{W}_t = \operatorname*{argmin}_{W \in \Omega} \frac{1}{\eta} B_\psi(W, W_t) + \langle \nabla L_t(W_t), W \rangle,$$
$$W_{t+1} = \operatorname*{argmin}_{W \in \Omega} \frac{1}{\eta} \{B_\psi(W, \tilde{W}_t) + r(W)\}.$$

Proof. The optimal solution to the first step satisfies, $\nabla\psi(\tilde{W}_t) - \nabla\psi(W_t) + \eta \nabla L_t(W_t) = 0$, so that

$$\nabla\psi(\tilde{W}_t) = \nabla\psi(W_t) - \eta \nabla L_t(W_t). \tag{4.22}$$

Then look at the optimal solution for the second step. For some $r'(W_{t+1}) \in \partial r(W_{t+1})$, we have

$$\nabla\psi(W_{t+1}) - \nabla\psi(\tilde{W}_t) + \eta r'(W_{t+1}) = 0. \tag{4.23}$$

Substitute Eq. (4.22) into Eq. (4.23), we obtain

$$\frac{1}{\eta}(\nabla\psi(W_{t+1}) - \nabla\psi(W_t)) + \nabla L_t(W_t) + r'(W_{t+1}) = 0,$$

which satisfies the optimal solution to the one-step update of (4.11).

We next show that ROMCO can achieve a sublinear regret in the following theory.

THEOREM 2. *The algorithm ROMCO (Alg. 1) runs over a sequential instances for each of m tasks. Assume that $r(0) = 0$, i.e., $W_1 = 0$ and $\max_t \|\nabla L_t(W_t)\| \leq G_2$, $U, V \in \mathbb{R}^{d \times m}$, then the following inequality holds for all $W^* \in \Omega$,*

$$\begin{aligned} R_\phi &\leq \frac{1}{2\eta} \|W^*\|_F^2 + T\eta G_2^2 \\ &= O(G_2 \|W^*\| \sqrt{T}). \end{aligned} \tag{4.24}$$

Proof. Let $\psi(\cdot) = \frac{1}{2}\|\cdot\|_F^2$, the solutions in subgradient projection (4.15) and (4.16) is equivalent to the one in form of the general optimization (4.11) according

to Lemma 4.4. According to the Lemma 4.3, given any $U^* \in \Omega$,

$$R_\phi \leq \frac{1}{\eta} B_\psi(W^*, W_1) + r(W_1) + \frac{\eta}{2\alpha} \sum_{t=1}^{T} \|\nabla L_t(W_t)\|_q^2.$$

Because that $\psi(\cdot) = \frac{1}{2}\|\cdot\|_F^2$ (i.e., $p = q = 2$), $\|\nabla L_t(W_t)\| \leq G_2$. Assume $W_1 = 0$, we obtain $r(W_1) = 0$ and $B_\psi(W^*, W_1) = \frac{1}{2}\|W^*\|_F^2$. Thus,

$$R_\phi \leq \frac{1}{2\eta}\|W^*\|_F^2 + \frac{T\eta}{2} G_2^2.$$

By setting $\eta = \frac{\|W^*\|}{\sqrt{T}G_2}$, we have $R_\phi = O(G_2\|W^*\|\sqrt{T})$.

5 Experimental Results

We evaluate the performance of our algorithm on three real-life datasets. We start by introducing the experimental data and benchmark setup, followed by discussion on the results.

5.1 Data and Benchmark Setup

5.1.1 Experimental Dataset We use three real-life datasets to evaluate our algorithm:

Spam Email[1]: A dataset hosted by the Internet Content Filtering Group contains 7068 emails collected from mailboxes of 4 users (i.e., 4 tasks). Each mail entry is converted to a word document vector using the TF-IDF representation. The task is to classify each email message into two categories: *legitimate* or *spam* for each user. Since the email dataset has no time-stamp, each email list is shuffled into a random sequence.

Human MHC-I[2]: It is a binary labeled human MHC-I dataset, containing 18664 peptide sequences for 12 MHC-I molecules. Each peptide sequence is converted to a 400 dimensional feature vector following [42]. The goal is to determine whether a peptide sequence (instance) is *binder* or *non-binder* to a MHC-I molecule (task). There are a total of 18664 samples with 400 features for 12 tasks.

EachMovie[3]: It contains 72916 users who rate a subset of 1628 movies. We randomly select 30 users (tasks) who viewed exactly 200 movies with their rating as target classes. Given each of 30 users, we then select 1783 users who viewed the same 200 movies and use their ratings as the feature of the movies. The six possible ratings (i.e., $\{1, \ldots, 6\}$) are converted into binary classes (i.e., *like* or *dislike*) based on the rating order. Finally we obtain 200 instances (1783 features) for each of 30 tasks. Tab. 1 summarizes the statistics of three datasets.

Table 1: Statistics of three datasets

	Spam Email	MHC-I	EachMovie
#Tasks	4	12	30
#Sample	7068	18664	6000
#Dimesion	1458	400	1783
#MaxSample	4129	3793	200
#MinSample	710	415	200

5.1.2 Baseline We compare our ROMCO algorithm with two batch learning methods: multi-task feature learning (*MTFL*) [43], and trace-norm regularized multi-task learning (*TRML*) [44], as well as three online learning algorithms: online multi-task learning (*OMTL*) [45], online passive-aggressive algorithm (*PA*) [46], and confidence-weighted online collaborative multi-task learning (*CW-COL)* [42]. Due to the high computational cost of the batch models, we modify MTFL and TRML to handle online data by periodically retraining them after observing 100 samples. All parameters for MTFL and TRML are set as default values. To further examine the effectiveness of learning multiple related tasks jointly, we make two variations of the PA algorithm as below: 1) *PA-Global*: It learns a single classification model from all tasks data. 2) *PA-Unique*: It trains a personalized classifier for each task using its own data. The parameter C is set to 1 for all PA variations and OMTL. All parameters for the CW-COL are tuned with a grid search on a held-out random shuffle. The four parameters η, τ, λ_1 and λ_2 for ROMCO are tuned by a grid search $\{10^{-7}, \ldots, 10^3\}$ on a held-out random shuffle as well.

5.1.3 Evaluation Metrics We evaluate the performance of aforementioned algorithms by two metrics: 1) Cumulative error rate: the ratio of predicted errors over a sequence of instances. It reflects the prediction accuracy of online learners. 2) F1-measure: the harmonic mean of precision and recall. It is suitable for evaluating the learner's performance on class-imbalanced datasets. We randomly shuffle the ordering of samples for each dataset, and repeat experiment 10 times with new shuffles. The average result and its standard deviation are reported below.

5.2 Comparison Result

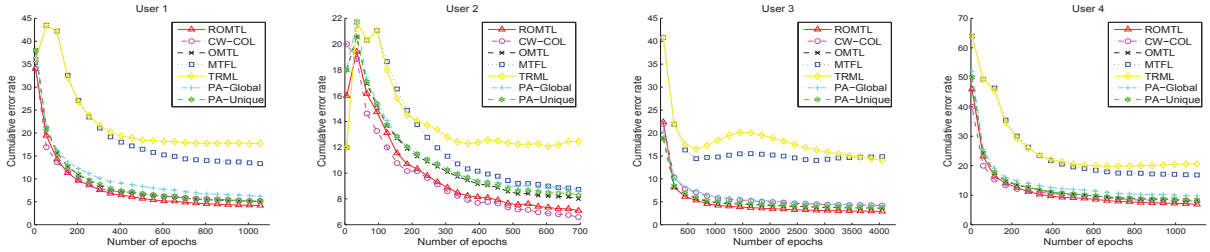
Tab. 2 shows the comparison results on three datasets in terms of average

[1] http://labs-repos.iit.demokritos.gr/skel/i-config/
[2] http://web.cs.iastate.edu/ honavar/ailab/
[3] http://goldberg.berkeley.edu/jester-data/

Table 2: Cumulative error rate (%) and F1-measure (%) on three datasets

	Spam Email		*MHC-I*		*EachMovie*	
	Error Rate	F1-measure	Error Rate	F1-measure	Error Rate	F1-measure
MTFL	13.40 (3.47)	88.39 (4.67)	43.84 (0.50)	51.04 (0.40)	27.51 (12.25)	79.18 (12.87)
TRML	16.21 (3.77)	86.02 (5.27)	44.26 (0.52)	50.50 (0.46)	26.58 (11.82)	79.89 (12.49)
PA-Global	7.08 (2.28)	94.04 (2.28)	44.70 (0.40)	45.44 (0.34)	31.80 (5.87)	74.43 (8.61)
PA-Unique	6.35 (2.04)	94.71 (1.70)	41.62 (0.21)	51.08 (0.28)	19.68 (7.39)	82.97 (9.35)
CW-COL	5.94 (1.67)	94.93 (1.93)	41.32 (0.37)	50.89 (0.49)	25.45 (6.96)	78.89 (9.30)
OMTL	6.20 (2.31)	94.79 (2.19)	41.56 (0.20)	51.13 (0.28)	19.44 (7.28)	83.18 (9.29)
ROMCO	**4.97 (2.00)**	**95.92 (1.55)**	**37.55 (0.24)**	**55.43 (0.31)**	**18.03 (6.57)**	**84.68 (8.39)**

Figure 1: Cumulative error rate on the Email Spam dataset along the entire online learning process



cumulative error rate and F1-measure (standard deviation is shown in brackets). Fig. 1 and Fig. 2 depict the detailed evolution of cumulative error rate along the entire online learning process of the Email Spam dataset and EachMovie dataset, respectively. In all figures, the horizontal-axis represents the online learning round, while the vertical-axis is the cumulative error rate or F1-measure, averaging over 10 times of shuffling order. In addition, the run-time of each algorithm, i.e., the time consumed during the whole online learning process, is shown in Tab. 3.

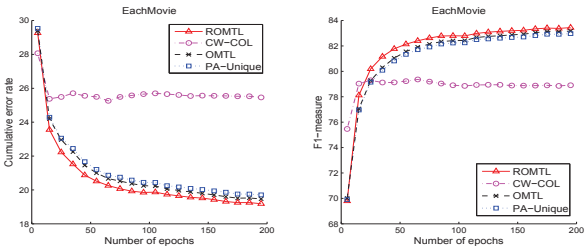
It can be seen that the proposed ROMCO outperforms others over all three datasets. In particular, the ROMCO always enjoys smaller error rates and higher F1-measures compared to other baselines. It shows that our algorithm can maintain a high quality of predicted accuracy. We believe that the good result is generally due to two reasons: First, the *personalized* and *correlative* patterns are effective to discover the personalized tasks and task relativeness, and these patterns are successfully captured in three real-world datasets. Second, once an error occurs from the m tasks, ROMCO would update the whole matrix. This would benefit other related tasks as the shared subspaces would be updated by the errors.

In addition, online methods (ROMCO and OMTL) are always better than that of the two batch learning algorithms (MTFL and TRML). Compared to online learner that conducts an update with current instance, offline learner updates with a substantial amount of samples. Consequently, ROMCO runs

Table 3: Run-time (in seconds) for each algorithm

	Spam Email	MHC-I	EachMovie
TRML	202.19	361.42	4804.25
MTFL	271.84	198.90	8548.12
PA-Global	0.86	1.79	11.27
PA-Unique	0.73	1.53	11.46
CW-COL	0.86	3.13	10.65
OMTL	24.62	42.92	50.01
ROMCO	11.49	11.45	17.92

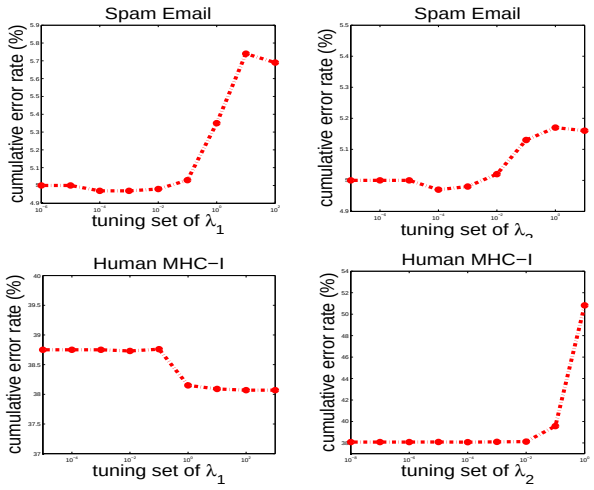
Figure 2: Average error rate F1-measure on the EachMovie dataset over 30 tasks along the entire online learning process



faster than offline methods as observed in Tab. 3.

Finally, we observe that ROMCO is slightly slower than CW-COL algorithm. This is expected as ROMCO has to update two component weight matrices. However, the extra computational cost is worth considered the significant improvement over two measurements has been achieved by using the two components.

Figure 3: Sensitivity study on the effect of the parameter λ_1 and λ_2 in terms of error rate



5.3 Effect of the Regularization Parameters

We use Spam Email and Human MHC-I dataset as the cases for parameter study. In Spam Email dataset, by fixing $\lambda_2 = 0.0001$ as well as varying the value of λ_1 in tuning set, i.e., $[10^{-6}, \ldots, 10^{2}]$, we study how the parameter λ_1 affects the classification performance of ROMCO; by fixing $\lambda_1 = 0.0001$ as well as varying the value of λ_2 in tuning set of $[10^{-7}, \ldots, 10]$, we study how the parameter affects the performance of ROMCO. Similarly, in Human MHC-I dataset, we study the pair of (λ_1, λ_2) by fixing $\lambda_2 = 0.0001$ with tuning set of λ_1 $[10^{-5}, \ldots, 10^{3}]$ and by fixing $\lambda_1 = 100$ with tuning set of λ_2 $[10^{-6}, \ldots, 10^{2}]$. In Fig. 3, we show the classification performance of ROMCO in terms of error rate for each pair of (λ_1, λ_2). From Fig. 3, we observe that the performance is worse with an increment of either λ_1 or λ_2 over Spam Email dataset. It indicates a weak relativeness among the tasks and many personalized tasks existing in Email dataset. In Human MHC-I, the bad performance is triggered by a small value of λ_1 or a large value of λ_2. Compared with Email data, MHC-I contains fewer personalized tasks, meanwhile most tasks are closely related and well represented by a low-dimensional subspace.

6 Conclusion

We propose an online MTL method which can identify sparse personalized patterns for outlier tasks, meanwhile captures a shared low-rank subspace for correlative tasks. As an online technique, the proposed algorithm can achieve a low prediction error rate via leveraging previous learned knowledge. As a multitask approach, it can balance the trade-off between the personalized model and the knowledge learned from other tasks. We show that it is able to achieve a sub-linear regret bound with respect to best linear model in hindsight. This can be regarded as a theoretical support for the proposed algorithm. Meanwhile, the empirical results demonstrate that our algorithms outperform other state-of-the-art techniques on three real-life applications.

References

[1] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[2] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *JMLR*, vol. 6, pp. 615–637, 2005.

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

[4] Y. Qi, O. Tastan, J. G. Carbonell, J. Klein-Seetharaman, and J. Weston, "Semi-supervised multi-task learning for predicting interactions between hiv-1 and human proteins," *Bioinformatics*, vol. 26, no. 18, pp. i645–i652, 2010.

[5] W. Pan, E. W. Xiang, N. N. Liu, and Q. Yang, "Transfer learning in collaborative filtering for sparsity reduction." in *AAAI*, vol. 10, 2010, pp. 230–235.

[6] S. C. H. Hoi, J. Wang, and P. Zhao, "LIBOL: a library for online learning algorithms," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 495–499, 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2627450

[7] A. Saha, P. Rai, H. Daumé III, and S. Venkatasubramanian, "Online learning of multiple tasks and their relationships," in *AISTATS*, Ft. Lauderdale, Florida, 2011.

[8] G. Lugosi, O. Papaspiliopoulos, and G. Stoltz, "Online multi-task learning with hard constraints," *arXiv preprint arXiv:0902.3526*, 2009.

[9] G. Cavallanti, N. Cesa-Bianchi, and C. Gentile, "Linear algorithms for online multitask classification," *JMLR*, vol. 11, pp. 2901–2934, 2010.

[10] P. Ruvolo and E. Eaton, "Online multi-task learning via sparse dictionary optimization," in *AAAI-14*, 2014.

[11] J. Attenberg, K. Weinberger, A. Dasgupta, A. Smola, and M. Zinkevich, "Collaborative email-spam filtering with the hashing trick," in *Proceedings of the Sixth Conference on Email and Anti-Spam*, 2009.

[12] P. Gong, J. Ye, and C. Zhang, "Robust multi-task feature learning," in *ACM SIGKDD (2012)*, 2012, pp. 895–903.

[13] J. Baxter, "A model of inductive bias learning," *J. Artif. Intell. Res.(JAIR)*, vol. 12, pp. 149–198, 2000.
[14] B. Bakker and T. Heskes, "Task clustering and gating for bayesian multitask learning," *JMLR*, vol. 4, pp. 83–99, 2003.
[15] K. Yu, V. Tresp, and A. Schwaighofer, "Learning gaussian processes from multiple tasks," in *ICML*. ACM, 2005, pp. 1012–1019.
[16] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *JMLR*, vol. 6, pp. 1817–1853, 2005.
[17] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *ACM SIGKDD (KDD04)*. New York, NY, USA: ACM, 2004, pp. 109–117.
[18] T. K. Pong, P. Tseng, S. Ji, and J. Ye, "Trace norm regularization: Reformulations, algorithms, and multi-task learning," *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 3465–3489, 2010.
[19] S. Negahban and M. J. Wainwright, "Estimation of (near) low-rank matrices with noise and high-dimensional scaling," *The Annals of Statistics*, pp. 1069–1097, 2011.
[20] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008.
[21] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient l 2, 1-norm minimization," in *UAI*. AUAI Press, 2009, pp. 339–348.
[22] X. Yang, S. Kim, and E. P. Xing, "Heterogeneous multitask learning with joint sparsity constraints," in *NIPS*, 2009, pp. 2151–2159.
[23] P. Zhao, S. C. H. Hoi, and R. Jin, "Double updating online learning," *JMLR*, vol. 12, pp. 1587–1615, 2011.
[24] P. Zhao, S. C. H. Hoi, R. Jin, and T. Yang, "Online AUC maximization," in *ICML-11*, 2011, pp. 233–240.
[25] P. Zhao and S. C. H. Hoi, "Cost-sensitive online active learning with application to malicious URL detection," in *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, 2013, pp. 919–927. [Online]. Available: http://doi.acm.org/10.1145/2487575.2487647
[26] O. Dekel, P. M. Long, and Y. Singer, "Online learning of multiple tasks with a shared loss," *JMLR*, vol. 8, pp. 2233–2264, October 2007.
[27] J. Abernethy, P. L. Bartlett, and A. Rakhlin, "Multitask learning with expert advice," in *COLT*, 2007, pp. 484–498.
[28] A. Agarwal, A. Rakhlin, and P. Bartlett, "Matrix regularization techniques for online multitask learning," EECS Department, University of California, Berkeley, Tech. Rep., Oct 2008.
[29] H. Cohen and K. Crammer, "Learning multiple tasks in parallel with a shared annotator," in *NIPS*, 2014, pp. 1170–1178.
[30] S. Shalev-Shwartz and A. Tewari, "Stochastic methods for l 1-regularized loss minimization," *JMLR*, vol. 12, pp. 1865–1892, 2011.
[31] S. Kim and E. P. Xing, "Tree-guided group lasso for multi-task regression with structured sparsity," in *ICML-10*, 2010, pp. 543–550.
[32] J. Chen, J. Liu, and J. Ye, "Learning incoherent sparse and low-rank patterns from multiple tasks," *TKDD*, vol. 5, no. 4, p. 22, 2012.
[33] D. P. Bertsekas, "Nonlinear programming," 1999.
[34] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM journal on control and optimization*, vol. 14, no. 5, pp. 877–898, 1976.
[35] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR computational mathematics and mathematical physics*, vol. 7, no. 3, pp. 200–217, 1967.
[36] A. Beck and M. Teboulle, "Mirror descent and nonlinear projected subgradient methods for convex optimization," *Operations Research Letters*, vol. 31, no. 3, pp. 167–175, 2003.
[37] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM review*, vol. 38, no. 1, pp. 49–95, 1996.
[38] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
[39] 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.
[40] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.
[41] J. C. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari, "Composite objective mirror descent." in *COLT*, 2010, pp. 14–26.
[42] G. Li, K. Chang, S. C. H. Hoi, W. Liu, and R. Jain, "Collaborative online learning of user generated content," in *CIKM*, 2011, pp. 285–290.
[43] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *NIPS*, 2006, pp. 41–48.
[44] J. Zhou, J. Chen, and J. Ye, *MALSAR: Multi-tAsk Learning via StructurAl Regularization*, Arizona State University, 2011.
[45] O. Dekel, P. M. Long, and Y. Singer, "Online multitask learning," in *COLT*, 2006, pp. 453–467.
[46] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *JMLR*, vol. 7, pp. 551–585, 2006.