Confidence Weighted Multitask Learning

Peng Yang,[1] **Peilin Zhao,**[2] **Jiayu Zhou,**[3] **Xin Gao**[1]
[1]King Abdullah University of Science and Technology, Saudi Arabia
[2]Tencent AI Lab, China, [3]Michigan State University, USA
{peng.yang.2, xin.gao}@kaust.edu.sa, masonzhao@tencent.com, jiayuz@msu.edu

Abstract

Traditional online multitask learning only utilizes the first-order information of the datastream. To remedy this issue, we propose a confidence weighted multitask learning algorithm, which maintains a Gaussian distribution over each task model to guide online learning process. The mean (covariance) of the Gaussian Distribution is a sum of a local component and a global component that is shared among all the tasks. In addition, this paper also addresses the challenge of active learning on the online multitask setting. Instead of requiring labels of all the instances, the proposed algorithm determines whether the learner should acquire a label by considering the confidence from its related tasks over label prediction. Theoretical results show the regret bounds can be significantly reduced. Empirical results demonstrate that the proposed algorithm is able to achieve promising learning efficacy, while simultaneously minimizing the labeling cost.

Introduction

Multitask learning (MTL) aims to enhance the overall generalization performance by learning the related knowledge of multiple tasks. Most existing works in multitask learning focus on how to take advantage of the task relationship, either by sharing model parameters via regularization techniques (Argyriou, Evgeniou, and Pontil 2008; Zhang and Yeung 2010; Yang, Zhao, and Gao 2017) or learning cross-task data directly (Crammer and Mansour 2012). This paper focuses on a specific multitask setting where tasks are allowed to query labels by interacting with other tasks for difficult cases, for example, recommending new products based on customers' preferences on old ones.

In a broad sense, there are two settings to learn multiple tasks together: 1) batch learning, where an entire training set is available to the learner before training; and 2) online learning, where the model is trained over the data streams. In contrast to batch learning, which often suffers from expensive re-training costs whenever new training data come, online learning avoids re-training and learns incrementally from sequential data, which is much more efficient (Hoi et al. 2018; Zhao et al. 2011). Online MTL (OMTL) has been intensively studied, where each task learner receives an instance on each round and then predicts its label. After that the true label is revealed and the learner updates the model as necessary. Previous studies (Dekel, Long, and Singer 2007; Yang and Zhao 2015; Murugesan et al. 2016) mostly utilized the first-order information of datastream. Few studies worked on second-order algorithms. To remedy this issue, we propose a confidence weighted MTL. The confidence estimated through a local-global Gaussian distribution over each task model can guide the direction and scale of parameter updates. Specifically, updates not only fix predicted errors but also increase confidence.

However, such method assumes that the true labels are readily provided for all the tasks, which is impractical in several settings where the tasks (e.g., minority languages, new products) naturally have very few data labels. Online active learning (OAL) addresses this concern by letting the learner decide whether to request the true label of the current instance or not. Most of OAL techniques decide a query by estimating a confidence towards current prediction (Cesa-Bianchi, Gentile, and Zaniboni 2006; Dekel, Gentile, and Sridharan 2012). In the multitask learning setting, one can further reduce the total number of required labels by interacting with related peer tasks. This paper proposes an active multitask learning, where the learner determines a query by considering the confidence over label prediction from related peer tasks. The key idea is that when a task model is uncertain about its prediction, it would consult its peers. Theoretical results show the regret bounds can be significantly reduced. Empirical results demonstrate that the proposed technique in such setting minimizes the learning errors and labeling cost simultaneously.

Related Work

Existing works on OMTL focus on how to take advantage of task relationship. To achieve this, Lugosi et al. (Lugosi, Papaspiliopoulos, and Stoltz 2009) imposed a hard constraint across the task model parameters. Agarwal et al. (Agarwal, Rakhlin, and Bartlett 2008) used a matrix regularization on model weights, like the nuclear norm (Ding et al. 2018). And Dekel et al. (Dekel, Long, and Singer 2007) learned the task relationship via a global loss function. However, all these works assume that the true label is provided to each instance.

OAL addresses this problem by making the decision on whether to query the label over the data streams (Cesa-Bianchi, Gentile, and Zaniboni 2006; Dekel, Gentile, and

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Sridharan 2012). It can be smoothly extended to the multi-task learning setting by applying active learning for each local task separately or several related tasks. Saha et al. (Saha et al. 2011) utilized the learned task-relationship matrix to query labels of the instances with more shared information. Murugesan et. al. (Murugesan and Carbonell 2017) determined a query based on the first-order information of the predicted margin. To save the labeling cost, our query decision is made by leveraging the Gaussian distribution of the predicted margin, which is more effective to capture the informative instances compared with prior solutions.

This work is motivated by the recent OAL techniques in the multitask setting (Saha et al. 2011; Murugesan and Carbonell 2017). A query is determined based on the peer tasks if the single task model is uncertain about its perdition. Our query strategy is different from them in two aspects: 1) query decision is made by consulting the local-global knowledge of multiple tasks; and 2) query confidence is estimated based on the Gaussian distribution of the predicted margin. Finally, our method is related to confidence weighted (CW) learning (Crammer, Dredze, and Pereira 2009). Different from (Li et al. 2014), we adapt the CW technique into a multi-task setting that performs active learning across the multiple tasks.

Algorithm

In this section, we first introduce an OMTL problem, then solve this problem with an active learning strategy.

Problem Setting

Suppose there are K tasks and each task k has a sequence of T_k training data. In this work, we consider an online binary classification problem for each task. Let $\{(\mathbf{x}_t^k, y_t^k)\}_{t=1}^{T_k}$ be a set of instance-label pairs for the task k where $\mathbf{x}_t^k \in \mathbb{R}^d$ is the t^{th} instance and $y_t^k \in \{\pm 1\}$ is its corresponding true label. Let $\{\mathbf{w}^k\}_{k\in[K]}$ be a set of arbitrary vectors where $\mathbf{w}^k \in \mathbb{R}^d$. Given a model weight $\mathbf{w}$, we denote $\ell_t(\mathbf{w})$ as its instantaneous loss and $L_T^k(\mathbf{w}) = \sum_t^{T_k} \ell_t(\mathbf{w})$ be its cumulative loss. The goal is to achieve a low regret compared with the best linear function. Formally, the regret of a model is given by

$$Regret = \sum_{t=1}^{T_k} \ell_t(\mathbf{w}_t) - \inf_{\mathbf{w}} L_T^k(\mathbf{w}).$$

The objective is to let the loss of the online algorithm converge to the loss of the best linear function $\mathbf{w}$.

Confidence Weighted Multitask Learning

We propose a local-global MTL framework where the local and global memory is introduced to store parts of the weight vector of each task, motivated by (Evgeniou and Pontil 2004). Formally, the task weight $\mathbf{w}_t^k$ is modeled in terms of local and global memories on round t,

$$\mathbf{w}_t^k = \mathbf{u}_t + \mathbf{v}_t^k, \tag{1}$$

where the global memory $\mathbf{u}$ captures the interdependent information across all tasks, while the local memory $\mathbf{v}^k$ learns the unique characteristic of a single task. When the offset $\mathbf{v}^k$ is 'small', it indicates that the tasks are similar to each other.

To better explore the second-order structure of parameter weights, motivated by (Crammer, Dredze, and Pereira 2009), we assume that a weight $\mathbf{w}$ follows a Gaussian distribution $\mathbf{w} \sim \mathcal{N}(\mu, \mathbf{\Sigma})$ with mean vector $\mu \in \mathbb{R}^d$ and covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$. The values μ_i and $\Sigma_{i,i}$ encode the model's knowledge and confidence towards the weight w_i, i.e., the smaller the value of $\Sigma_{i,i}$ is, the more confident the learner is towards the mean value μ_i. The covariance term $\Sigma_{i,j}$ captures the interactions between w_i and w_j. To adapt confidence weight into the local-global setting in Eq. (1), we begin with the following Lemma:

Lemma 1. *If* $\mathbf{u} \sim \mathcal{N}(\mathbf{p}, \mathbf{A})$ *and* $\{\mathbf{v}^k \sim \mathcal{N}(\mathbf{q}^k, \mathbf{B}^k)\}_{k=1}^K$ *are mutually independent normal random variables, then the linear combination:* $\mathbf{w}^k = \mathbf{u} + \mathbf{v}^k$ *follows the Gaussian distribution:*

$$\mathbf{w}^k \sim \mathcal{N}\left(\mathbf{p} + \mathbf{q}^k, \mathbf{A} + \mathbf{B}^k\right).$$

Proof. The proof is in the Supplementary Material[1]. □

Given an instance $(\mathbf{x}_t^k, y_t^k)$ at round t, the local-global parameters aim to adjust their distributions to ensure the probability of a correct prediction at least $\eta > 0$:

$$\begin{aligned}\min_{\mathbf{p},\mathbf{A},\mathbf{q}^k,\mathbf{B}^k} \sum_{k=1}^{K} &\mathbb{D}_{KL}\left(\mathcal{N}(\mathbf{q}^k, \mathbf{B}^k)||\mathcal{N}(\mathbf{q}_{t-1}^k, \mathbf{B}_{t-1}^k)\right) \\ &+ \mathbb{D}_{KL}\left(\mathcal{N}(\mathbf{p}, \mathbf{A})||\mathcal{N}(\mathbf{p}_{t-1}, \mathbf{A}_{t-1})\right), \\ \text{s.t. } &\Pr_{\mathbf{w}^k\sim\mathcal{N}(\mathbf{p}+\mathbf{q}^k,\mathbf{A}+\mathbf{B}^k)}[y_t^k(\mathbf{w}^k \cdot \mathbf{x}_t^k) \geq 0] \geq \eta\end{aligned}$$

where $\mathbb{D}_{KL}$ is the Kullback-Leibler divergence:

$$\begin{aligned}&\mathbb{D}_{KL}(\mathcal{N}(\mu, \mathbf{\Sigma})||\mathcal{N}(\mu_t, \mathbf{\Sigma}_t)) \\ =&\frac{1}{2}\left(\log\left(\frac{|\mathbf{\Sigma}_t|}{|\mathbf{\Sigma}|}\right) + \text{Tr}\left(\frac{\mathbf{\Sigma}}{\mathbf{\Sigma}_t}\right) + \|\mu_t - \mu\|^2_{\mathbf{\Sigma}_t^{-1}} - d\right),\end{aligned}$$

$|\ |$ is the determinant of a matrix and Tr() is the trace of a matrix. In the following Lemma, The constraint can be formulated by the Gaussian distribution.

Lemma 2. *The predicted margin on* $(\mathbf{x}^k, y^k)$ *by the model* $\mathbf{w}^k \sim \mathcal{N}(\mathbf{p} + \mathbf{q}^k, \mathbf{A} + \mathbf{B}^k)$ *follows the Gaussian distribution:*

$$y^k(\mathbf{w}^k \cdot \mathbf{x}^k) \sim \mathcal{N}\left(y^k((\mathbf{p} + \mathbf{q}^k) \cdot \mathbf{x}^k),\ \mathbf{x}^\top(\mathbf{A} + \mathbf{B}^k)\mathbf{x}\right).$$

The probability constraint can be written explicitly as:

$$-y^k((\mathbf{p} + \mathbf{q}^k) \cdot \mathbf{x}^k) + \phi\sqrt{\mathbf{x}^{k\top}(\mathbf{A} + \mathbf{B}^k)\mathbf{x}^k} \leq 0,$$

where $\phi = \Phi^{-1}(\eta) > 0$ *and* Φ *is the Gaussian cumulative distribution function.*

Proof. The proof is in the Supplementary Material[1]. □

We directly tackle the variance variable and the problem becomes convex, while replacing $-y^k((\mathbf{p}+\mathbf{q}^k)\cdot\mathbf{x}^k$ with the

[1]https://github.com/YoungBigBird1985/Second-Order-Online-Multitask-Learning

hinge loss $\ell(\mathbf{p}+\mathbf{q}^k) = \max(0, 1-y^k((\mathbf{p}+\mathbf{q}^k)\cdot\mathbf{x}^k))$. We recast the the constraint as a regularizer, solving the following unconstrained function:

$$\begin{aligned}\min_{\mathbf{p},\mathbf{A},\mathbf{q}^k,\mathbf{B}^k} & \sum_{k=1}^{K} \mathbb{D}_{KL}\left(\mathcal{N}(\mathbf{q}^k,\mathbf{B}^k)||\mathcal{N}(\mathbf{q}^k_{t-1},\mathbf{B}^k_{t-1})\right) \\ & + \mathbb{D}_{KL}\left(\mathcal{N}(\mathbf{p},\mathbf{A})||\mathcal{N}(\mathbf{p}_{t-1},\mathbf{A}_{t-1})\right) \\ & + \sum_{k=1}^{K}\left(\frac{1}{2\epsilon}\ell^t(\mathbf{p}+\mathbf{q}^k) + \frac{1}{2\lambda}\mathbf{x}_t^{k\top}(\mathbf{A}+\mathbf{B}^k)\mathbf{x}_t^k\right),\end{aligned} \tag{2}$$

where ϵ and λ are positive tradeoff parameters. Specifically, the objective (2) aims to reach the trade-off between the distribution divergence (the first two terms), the loss function (the third term) and the predicted variance (the fourth term). In other words, the objective tends to make the least adjustment to minimize the painful loss and maximize the confidence of prediction. To solve this problem, we exploit the block coordinate descent (Tseng 2001) to optimize the local-global memory variables alternatively.

Optimizing Local Memory The parameters $\mathbf{B}^k$ can be solved as below,

$$f(\mathbf{B}^k) = \log\left(\frac{|\mathbf{B}^k_{t-1}|}{|\mathbf{B}^k|}\right) + \mathrm{Tr}\left(\frac{\mathbf{B}^k}{\mathbf{B}^k_{t-1}}\right) + \frac{1}{\lambda}\mathbf{x}_t^{k\top}\mathbf{B}^k\mathbf{x}_t^k.$$

By applying the KKT condition on $\mathbf{B}$, we have that $(\mathbf{B}^k_t)^{-1} = (\mathbf{B}^k_{t-1})^{-1} + \frac{1}{\lambda}\mathbf{x}_t\mathbf{x}_t^\top$. By using the Sherman–Morrison formula (Sherman and Morrison 1950), $\mathbf{B}^k_t$ can be updated efficiently with time complexity $O(d^2)$,

$$\mathbf{B}^k_t = \mathbf{B}^k_{t-1} - \frac{\mathbf{B}^k_{t-1}\mathbf{x}^k_t\mathbf{x}^{k\top}_t\mathbf{B}^k_{t-1}}{\lambda + \mathbf{x}^{k\top}_t\mathbf{B}^k_{t-1}\mathbf{x}^k_t}. \tag{3}$$

Let $\mathbf{p} = \mathbf{p}_{t-1}$, the $\mathbf{q}^k$ is solved under the hinge loss and squared hinge loss, respectively.

Lemma 3. *Let $\hat{e}^k_t = (\mathbf{p}_{t-1}+\mathbf{q}^k_{t-1})\cdot\mathbf{x}^k_t$. Whenever $y^k_t \neq \mathrm{sign}(\hat{f}^k_t)$, we solve the problem*

$$f(\mathbf{q}^k) = \|\mathbf{q}^k - \mathbf{q}^k_{t-1}\|^2_{(\mathbf{B}^k_t)^{-1}} + \frac{1}{\epsilon}\ell_t(\mathbf{p}_{t-1}+\mathbf{q}^k),$$

where the optimal solution of $\mathbf{q}^k$ is given by,

$$\mathbf{q}^k_t = \mathbf{q}^k_{t-1} + g^k_t y^k_t \mathbf{B}^k_t\mathbf{x}^k_t, \tag{4}$$

where

$$g^k_t = \frac{\max\{0, 1-y^k_t\hat{e}^k_t\}}{\epsilon + \mathbf{x}^{k\top}_t\mathbf{B}^k_t\mathbf{x}^k_t} \quad \textit{(squared hinge)}$$

$$g^k_t = \min\left\{\frac{1}{2\epsilon}, \max\left\{0, \frac{1-y^k_t\hat{e}^k_t}{\mathbf{x}^{k\top}_t\mathbf{B}^k_t\mathbf{x}^k_t}\right\}\right\} \quad \textit{(hinge)}$$

Proof. The proof is in the supplementary material[1]. □

Optimizing Global Memory The global parameter $\mathbf{A}$ can be optimized:

$$f(\mathbf{A}) = \log\left(\frac{|\mathbf{A}_{t-1}|}{|\mathbf{A}|}\right) + \mathrm{Tr}\left(\frac{\mathbf{A}}{\mathbf{A}_{t-1}}\right) + \frac{1}{\lambda}\mathrm{Tr}\left(\mathbf{X}_t^\top\mathbf{A}\mathbf{X}_t\right),$$

Algorithm 1 CWMT: Confidence Weighted Multitask Learning

Input: $\lambda, \epsilon > 0$.
Output: $\mathbf{p}_T$, $\mathbf{A}_T$, $\mathbf{q}^k_T$, $\mathbf{B}^k_T, k \in [K]$.
Initialize: $\mathbf{p}_0 = \mathbf{q}^k_0 = \mathbf{0}, \mathbf{A}_0 = \mathbf{B}^k_0 = \mathbf{I},\ k \in [K]$.
for $t = 1, \ldots, T$ **do**
  for (local update): $k = 1, \ldots, K$ **in parallel do**
    Receive $\mathbf{x}^k_t$ and let $\mu^k_{t-1} = \mathbf{p}_{t-1} + \mathbf{q}^k_{t-1}$;
    Compute $\hat{y}^k_t = \mathrm{sign}(\mu^k_{t-1}\cdot\mathbf{x}^k_t)$;
    If $y^k_t \neq \hat{y}^k_t$, update $\mathbf{B}^k_t$ in Eq. (3), $\mathbf{q}^k_t$ in Eq. (4);
  endfor
  Reduce (global update): aggregate $\{z^1_t, \ldots, z^K_t\}$
  Update $\mathbf{A}_t$ with Eq. (5) and $\mathbf{p}_t$ with Eq. (6);
end for

where $\mathbf{X}_t = [\mathbf{x}^1_t, \mathbf{x}^2_t, \ldots, \mathbf{x}^K_t] \in \mathbb{R}^{d\times K}$. By using Woodbury matrix identity, $\mathbf{A}$ can be updated by

$$\mathbf{A}_t = \mathbf{A}_{t-1} - \mathbf{A}_{t-1}\mathbf{X}_t\mathbf{C}^{-1}_{t-1}\mathbf{X}^\top_t\mathbf{A}_{t-1}, \tag{5}$$

where $\mathbf{C}_{t-1} = \lambda\mathbf{I}_K + \mathbf{X}^\top_t\mathbf{A}_{t-1}\mathbf{X}_t$ is positive-definite and $\mathbf{I}_K \in \mathbb{R}^{K\times K}$ is an identity matrix. The matrix inverse in Eq. (5) takes $O(K^3+d^2K)$ complexity, which is acceptable when the task number K is small.

Let $z^k_t = \mathcal{I}(y^k_t \neq \hat{y}^k_t)$ where $\mathcal{I}(\cdot)$ is an indicator function, $\mathbf{p}$ is solved by

$$f(\mathbf{p}) = \|\mathbf{p} - \mathbf{p}_{t-1}\|^2_{\mathbf{A}^{-1}_t} + \frac{1}{\epsilon}\sum_{k=1}^{K} z^k_t\ell_t(\mathbf{p}+\mathbf{q}^k_{t-1}).$$

Taking the derivative of the above problem, i.e. $\nabla_{\mathbf{p}_{t-1}}f(\mathbf{p})$, $\mathbf{p}$ is solved by

$$\mathbf{p}_t = \mathbf{p}_{t-1} + \frac{1}{2\epsilon}\mathbf{A}_t\sum_{k=1}^{K} z^k_t y^k_t\mathbf{x}^k_t. \tag{6}$$

We summarize the confidence weighted multitask learning in Algorithm 1. It uses a conservative strategy to update the model when an error occurs. Different from first-order techniques (Murugesan et al. 2016), this algorithm captures the second-order information by exploiting the confidence of weight parameters over Gaussian distribution. Unlike the CW-based method (Li et al. 2014), we provide a theoretical analysis for the CW-based MTL in Theorem 1.

Theorem 1. *Let $\{(\mathbf{x}^k_t, y^k_t)\}^T_{t=1}$ be a sequence of samples on any task ($k \leq K$), where $\mathbf{x}^k_t \in \mathbb{R}^d$, $y^k_t \in \{\pm 1\}$. For any $\mu \in \mathbb{R}^d$ on the convex loss $\ell(\mu)$, the CWMT satisfies:*

$$Regret \leq \frac{\lambda\log(1+KT)}{4\epsilon} + \epsilon(\mathcal{D}(\mu))^2 Tr\left((\mathbf{A}_T + \mathbf{B}^k_T)^{-1}\right)$$

where $\mathcal{D}(\mu) = \max_t \|(\mathbf{p}_t + \mathbf{q}^k_t) - \mu\|$.

Proof. The proof is in the Supplementary Material[1]. □

Discussion: Setting $\epsilon = \frac{1}{2}\sqrt{\frac{\lambda\log(1+KT)}{(\mathcal{D}(\mu))^2\mathrm{Tr}\left((\mathbf{A}_T+\mathbf{B}^k_T)^{-1}\right)}}$, we get

$$Regret \leq \frac{1}{2}\sqrt{\lambda}\mathcal{D}(\mu)\sqrt{\mathrm{Tr}((\mathbf{A}_T + \mathbf{B}^k_T)^{-1})\log(1+KT)}.$$

Let $\mathbf{\Sigma}^k = \mathbf{A} + \mathbf{B}^k$, $\mathbf{A} \prec \mathbf{\Sigma}^k$ and $\mathbf{B}^k \prec \mathbf{\Sigma}^k$ yield to $\mathbf{A}^{-1} \succ (\mathbf{\Sigma}^k)^{-1}$ and $(\mathbf{B}^k)^{-1} \succ (\mathbf{\Sigma}^k)^{-1}$. Assume $\|\mathbf{x}_t\|_2 \leq 1$, we have $Tr((\mathbf{\Sigma}^k)^{-1}) \leq \frac{1}{2}Tr(\mathbf{A}^{-1}) + \frac{1}{2}Tr((\mathbf{B}^k)^{-1}) \leq O(\frac{(K+1)T}{\lambda})$. Thus, the regret is in the order of $O(\sqrt{KT})$.

Remark: As $Regret = \sum_{t=1}^{T_k} \ell_{hinge}^t(\mathbf{w}_t^k) - \inf_{\mathbf{w}} L_T^k(\mathbf{w})$, and $\sum_t \ell_{hinge}(\cdot) \geq \sum_t \ell_{0-1}(\cdot) = |M|$, then $regret \geq |M| - \inf_{\mathbf{w}} L_T^k(\mathbf{w})$. It infers a relative mistake bound:

$$\begin{aligned} &\mathbb{E}[M] - \inf_{\mathbf{w}} L_T^k(\mathbf{w}) \\ \leq &\frac{\lambda}{4\epsilon}\log(1+KT) + \epsilon(\mathcal{D}(\mu))^2 \mathrm{Tr}\left((\mathbf{A}_T + \mathbf{B}_T^k)^{-1}\right). \end{aligned} \tag{7}$$

Active Confidence Weighted Multitask Learning

Unlike the online algorithm that retrieves the label of every instance, active learning considers the labeling budget and has to decide whether to query the true label of the current instance $\mathbf{x}_t$. If the label is queried, the algorithm can update the learner with y_t; otherwise, no action is performed and the learner continues to process the next one. Query and update decisions are defined as binary variables Q_t and Z_t at time t, where $Q_t = 1$ when y_t is queried of, and 0 otherwise; Z_t is under the same setting. In general, optimizing the local memory $(\mathbf{B}, \mathbf{q})$ with query and update decisions on round t yields

$$\mathbf{B}_t^{-1} = \mathbf{B}_{t-1}^{-1} + Q_t Z_t \frac{\mathbf{x}_t \mathbf{x}_t^\top}{\lambda},\ \mathbf{q}_t = \mathbf{q}_{t-1} + Q_t Z_t g_t y_t \mathbf{B}_t \mathbf{x}_t.$$

The goal of active learning is to achieve few mistakes with a small query number $\sum_t Q_t$. Next we propose a query method to perform active leaning across multiple tasks.

Adaptive-Margin Query The target of the active learning algorithm is to make few mistakes with a small amount of queries. To achieve this goal, we propose a randomized query strategy based on a confidence score Θ_t: given $h > 0$, the binary variable Q_t is sampled using a Bernoulli distribution with a parameter $h/(h + \max(0, \Theta_t))$; if $Q_t = 1$, then the actual label y_t is queried of; otherwise no query is performed. The randomized query strategy has been studied in (Dekel, Gentile, and Sridharan 2012). We extend the randomized query into the multitask setting by performing active leaning across multiple tasks. We begin with the additional annotations below,

$$a_t^k = \mathbf{x}_t^{k\top} \mathbf{A}_{t-1} \mathbf{x}_t^k,\ \ b_t^k = \mathbf{x}_t^{k\top} \mathbf{B}_{t-1}^k \mathbf{x}_t^k.$$

Using the Sherman-Morrison formula, it is inferred that

$$\mathbf{x}_t^{k\top} \mathbf{\Sigma}_t^k \mathbf{x}_t^k = \mathbf{x}_t^{k\top} \mathbf{A}_t \mathbf{x}_t^k + \mathbf{x}_t^{k\top} \mathbf{B}_t^k \mathbf{x}_t^k,$$

where the last two terms can be derived as:

$$\mathbf{x}_t^{k\top}\left(\mathbf{A}_{t-1} - \frac{\mathbf{A}_{t-1}\mathbf{x}_t^k \mathbf{x}_t^{k\top} \mathbf{A}_{t-1}}{\lambda + \mathbf{x}_t^{k\top} \mathbf{A}_{t-1} \mathbf{x}_t^k}\right)\mathbf{x}_t^k = \frac{a_t^k}{1 + a_t^k/\lambda},$$

$$\mathbf{x}_t^{k\top}\left(\mathbf{B}_{t-1}^k - \frac{\mathbf{B}_{t-1}^k \mathbf{x}_t^k \mathbf{x}_t^{k\top} \mathbf{B}_{t-1}^k}{\lambda + \mathbf{x}_t^{k\top} \mathbf{B}_{t-1}^k \mathbf{x}_t^k}\right)\mathbf{x}_t^k = \frac{b_t^k}{1 + b_t^k/\lambda}.$$

Definition 1. *Assume a task weight $\mathbf{w}^k \sim \mathcal{N}(\mu^k, \mathbf{\Sigma}^k)$, with mean $\mu^k = \mathbf{p} + \mathbf{q}^k$ and variance $\mathbf{\Sigma}^k = \mathbf{A} + \mathbf{B}^k$. At round t, the algorithm decides to query with a probability $\frac{h}{h+\max(0,\Theta_t^k)}$ $(h > 0)$, where the score Θ_t^k is defined as,*

$$\Theta_t^k = |\Delta_t^k| - \frac{1}{4\epsilon} C_t^k, \tag{8}$$

where $|\cdot|$ is the absolute value,

$$\Delta_t^k = (\mathbf{p}_{t-1} + \mathbf{q}_{t-1}^k) \cdot \mathbf{x}_t^k;\ \ C_t^k = \frac{a_t^k}{1 + a_t^k/\lambda} + \frac{b_t^k}{1 + b_t^k/\lambda}.$$

Θ_t is a function parameterized by two variables $|\Delta_t|$ and $\mathcal{C}_t$. The variable $|\Delta_t|$ is a distance of an input to the decision boundary, known as the 'margin' (Cesa-Bianchi, Gentile, and Zaniboni 2006), while $\mathcal{C}_t$ is the confidence of the current prediction, known as 'variance'. In this way, $|\Delta_t| - \frac{1}{4\epsilon}\mathcal{C}_t$ acts as a lower confidence bound of the prediction. The probability of query $(h/(h + \max(0, \Theta_t)))$ will be reduced only when an input is far from the boundary (i.e. a large $|\Delta_t|$), meanwhile it has a low variance towards the prediction (i.e. $\mathcal{C}_t$). So that the predicted result is reliable, and a label can be omitted safely. Similar with (Murugesan and Carbonell 2017), the confidence Θ_t is estimated by exploiting the local-global knowledge of multiple tasks. The idea is that when the single task is uncertain about its prediction, it can consult the global memory that encodes the knowledge of similar tasks to help make the query decision.

Intuitively, a query decision is effective only if it can control the probability of making a mistake when the label is not queried. To analyze the effectiveness of the query method, we derive an error bound for our approach that learns from only queried labels $\{t : Q_t = 1\}$. For the randomized queries, the mistake trials can be partitioned into two disjoint sets: a set $\mathcal{S} = \{t : \frac{h}{h+\max(0,\Theta_t)} < 1\}$ includes indices on which a stochastic query is conducted, and a set $\mathcal{D} = \{t : \frac{h}{h+\max(0,\Theta_t)} = 1\}$ includes indices when there is a deterministic query. The expected number of queries (i.e., expected labeling cost) is upper bounded by

$$\mathbb{E}\left[|\mathcal{D}| + \sum_{t \in \mathcal{S}} \frac{h}{h + \Theta_t}\right].$$

Let $Z_t = 1$ if a mistake occurs at the round t, i.e., $\hat{y}_t \neq y_t$ after the true label is queried. We denote $\mathcal{M} = \{t : y_t \Delta_t \leq 0\}$ as the set of mistake trials and let $M = |\mathcal{M}|$, while $U_T = \{t : Z_t Q_t = 1, t \in [T]\}$ as the set of update trials.

Theorem 2. *Assume that $\{(\mathbf{x}_t^k, y_t^k)\}_{t=1}^T$ is a sequence of instances for any task k ($k \leq K$), where $\mathbf{x}_t^k \in \mathbb{R}^d$, $y_t^k \in \{\pm 1\}$. CWMT learns from the queried labels $\{t : Q_t^k = 1, s.t.\ Q_t^k \sim \frac{h}{h+\max(0,\Theta_t^k)}\}$. For any $\mu \in \mathbb{R}^d$, it satisfies*

$$\begin{aligned} \mathbb{E}[M] \leq &\sum_{t=1}^{T} \ell_t(\mathbf{p} + \mathbf{q}^k) + \frac{\lambda}{4h\epsilon}\log(1+KT) \\ &+ \frac{\epsilon}{h}\mathbb{E}\left[(\mathcal{D}(\mu))^2 Tr\left((\mathbf{A}_{U_T} + \mathbf{B}_{U_T}^k)^{-1}\right)\right], \end{aligned} \tag{9}$$

where $\mathcal{D}(\mu) = \max_t \|(\mathbf{p}_t + \mathbf{q}_t^k) - h\mu\|$.

Proof. The proof is in the Supplementary Material[1]. □

Algorithm 2 ACWMT: Active Confidence Weighted Multitask Learning

Input: $\lambda, \epsilon > 0$ and $h > 0$.
Output: $\mathbf{p}_T$, $\mathbf{A}_T$, $\mathbf{q}_T^k$, $\mathbf{B}_T^k, k \in [K]$.
Initialize: $\mathbf{p}_0 = \mathbf{q}_0^k = \mathbf{0}, \mathbf{A}_0 = \mathbf{B}_0^k = \mathbf{I}\ \forall k \in [K]$.
for $t = 1, \ldots, T$ **do**
  for (local update): $k = 1, \ldots, K$ **in parallel do**
    Receive $\mathbf{x}_t^k$ and $\hat{y}_t^k = \text{sign}(\mu_{t-1}^k \cdot \mathbf{x}_t^k)$;
    Compute Θ_t^k with Eq. (8);
    if $\Theta_t^k \leq 0$ **then (*aggressive update*)**
      Set $Q_t^k Z_t^k = 1$ and query the true label;
    else (*stochastic update*)
      Generate $Q_t^k \sim \frac{h}{h+\Theta_t^k}$;
      Query if $Q_t^k = 1$, let $Z_t^k = 1$ if $y_t^k \neq \hat{y}_t^k$;
    end if
    If $Q_t^k Z_t^k = 1$, update $\mathbf{B}_t^k$ in Eq. (3), $\mathbf{q}_t^k$ in Eq. (4);
  endfor
  Reduce (global update): aggregate $\{Z_t^1, \ldots, Z_t^K\}$
  Update $\mathbf{A}_t$ with Eq. (5) and $\mathbf{p}_t$ with Eq. (6);
end for

Discussion: We can replace $\sum_{t=1}^T \ell_t(\mathbf{p} + \mathbf{q}^k)$ with its loss of the best model, $\inf_\mu \sum_{t=1}^T \ell_t(\mu)$ where $\mu = \mathbf{p} + \mathbf{q}^k$, then Eq. (9) becomes a relative mistake bound (Abernethy, Bartlett, and Rakhlin 2007). It is observed that CWMT learned on actively selected labels can achieve a comparable mistake bound with the one learned on all labels in Eq. (7). It demonstrates the efficacy of the proposed query strategy.

Adaptive Optimization We solve the active multitask learning problem with adaptive optimization. The algorithm maintains $(\mathbf{B}^k, \mathbf{q}^k)$ for each local memory and $(\mathbf{A}, \mathbf{p})$ for the global memory. At round t, the learner receives an input $\mathbf{x}_t^k$ and predicts its binary label $\hat{y}_t^k = \text{sign}(\mu_{t-1}^k \cdot \mathbf{x}_t^k)$. Then its actual label y_t^k is revealed with a probability of $\frac{h}{h+\max(0,\Theta_t^k)}$, which yields a *stochastic query* or a *deterministic query*. In a stochastic query ($\frac{h}{h+\max(0,\Theta_t^k)} < 1$), update is driven by mistake. If an error occurs ($\hat{y}_t^k \neq y_t^k$), the algorithm updates the local memory in a recursive way; otherwise, no action is performed and we proceed to the next one with $\mathbf{B}_t^k = \mathbf{B}_{t-1}^k$ and $\mathbf{q}_t^k = \mathbf{q}_{t-1}^k$. We observe that a deterministic query is issued when $\frac{h}{h+\max(0,\Theta_t^k)} = 1$. In this case, *aggressive update* is performed, that is, we update a model even if no error occurs. After that, the global update $(\mathbf{p}, \mathbf{A})$ is conducted via aggregating the update decisions $\{Z^i\}_{i=1}^K$ from the local tasks. We summarize this algorithm ACWMT, active confidence weighted multitask learning, in Algorithm 2.

To further understand this algorithm, we compute under what condition an update will be issued aggressively. An aggressive update is issued when $\Theta_t^k \leq 0$, which yields

$$|\Delta_t^k| \leq \theta_t^k(a,b) = \frac{1}{4\epsilon}\left(\frac{a_t^k}{1 + a_t^k/\lambda} + \frac{b_t^k}{1 + b_t^k/\lambda}\right).$$

If $|\Delta_t^k|$ is less than $\theta_t^k(a,b)$, a deterministic query/update is conducted, while $|\Delta_t^k|$ is above $\theta_t^k(a,b)$, a label is queried with a probability less than 1. And the upper bound of $\theta_t^k(a,b)$ increases with a or b. Since $\mathbf{0} \preceq \mathbf{A}_t, \mathbf{B}_t \preceq \mathbf{I}$ and $\|\mathbf{x}_t\| \leq 1$, it is inferred that $0 \leq a_t^k, b_t^k \leq 1$. When $a_t^k, b_t^k = 0$ with a minimal uncertainty, a deterministic query is issued only if the input lies on the boundary ($|\Delta_t^k| \leq \theta_t^k(a,b) = 0$). When $a_t^k, b_t^k = 1$ with the largest value of uncertainty, this implies that an aggressive query is issued whenever its margin $|\Delta_t^k|$ is less than $\theta_t^k(a,b) = \frac{\lambda}{2\epsilon(\lambda+1)}$.

In the following analysis, we show the superiority of ACWMT. Besides stochastic query trials $\mathcal{S}$ and deterministic query trials $\mathcal{D}$, we denote $\mathcal{V} = \{t : y_t\Delta_t > 0, \Theta_t \leq 0\}$ as the set of trials where there is an aggressive update without predicted errors, and let $V = |\mathcal{V}|$.

Theorem 3. *Let $\{(\mathbf{x}_t^k, y_t^k)\}_{t=1}^T$ be an input-label pair sequence for any task k, $k \in [K]$. Following the setting in Theorem 2, the proposed ACWMT satisfies*

$$\mathbb{E}[M] \leq \sum_{t=1}^T \ell(\mu; \mathbf{x}_t^k, y_t^k) + \frac{\lambda}{4h\epsilon}\log(1 + KT) + \frac{\epsilon}{h}\mathbb{E}\left[(\mathcal{D}(\mu))^2 Tr\left((\mathbf{A}_{U_T} + \mathbf{B}_{U_T}^k)^{-1}\right)\right] - \mathbb{E}[V].$$

The expected number of update is $\mathbb{E}[|\mathcal{D}| + \sum_{t\in\mathcal{S}\cap\mathcal{M}} \frac{h}{h+\Theta_t}]$.

Proof. The proof is Supplementary Material[1]. □

Discussion: It is observed that the error bound of ACWMT, in expectation, is lower than that of the conservative update algorithm in Theorem 2, due to the deduction of $\mathbb{E}[V]$ from the bound, where $\mathbb{E}[V]$ is the expected number of the aggressive update trials. This can be regarded as a theoretical support for our aggressive algorithm. Although the aggressive strategy requires more updates in an early stage, it helps reduce the predicted variance and accelerate the learning progress, which could reduce the error rate and queried number when the model learns sufficient knowledge of data.

Table 1: Description of the data sets

	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

Experiments

We conduct the performance evaluation for the algorithms on three real-world datasets. We begin with the introduction of the experimental data and evaluation metrics. Then we show and discuss the empirical results.

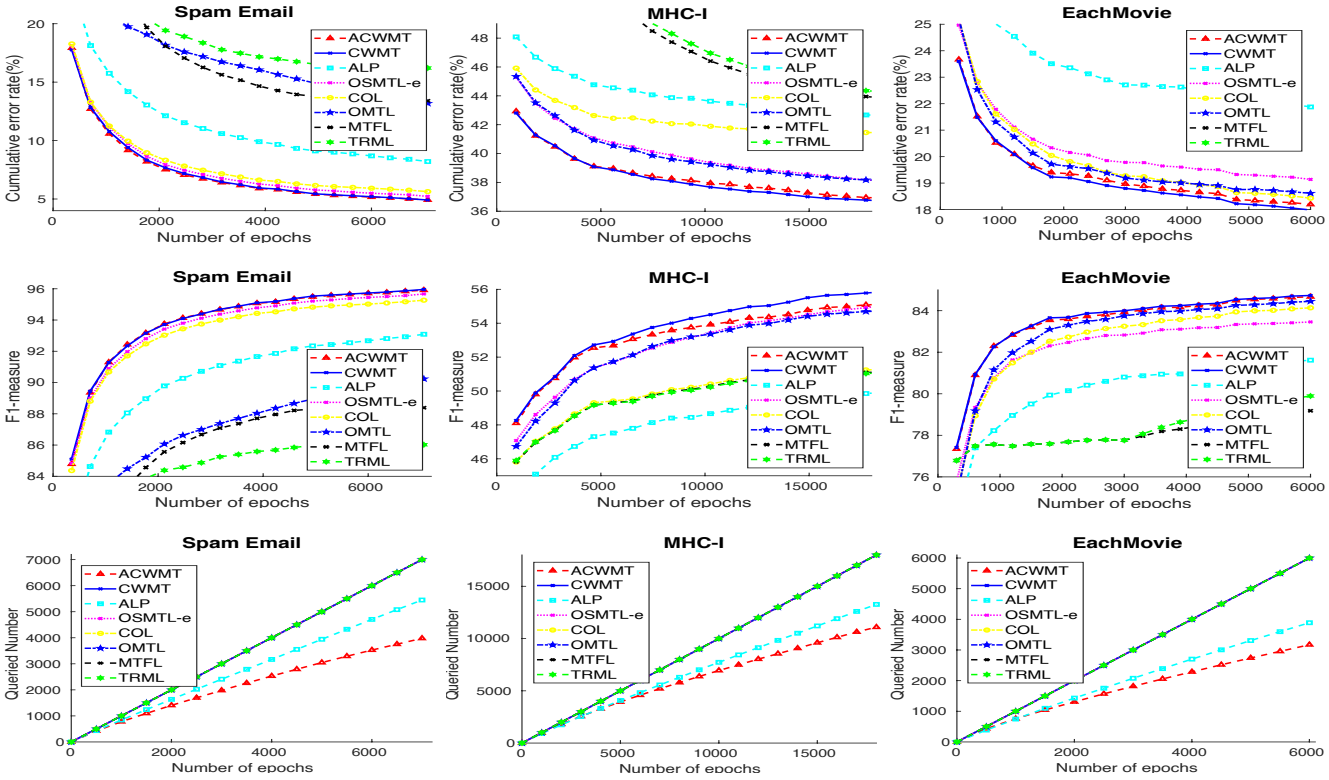
Experimental Datasets

We introduce three real-world datasets to evaluate the methods. Table 1 summarizes the statistic features of the datasets.

Table 2: Mean and standard deviation of cumulative error rate (%), F1-measure (%) and queried number on the datasets

Algorithm	*Spam Email*			*MHC-I*			*EachMovie*		
	Error Rate	F1-measure	Query	Error Rate	F1-measure	Query	Error Rate	F1-measure	Query
MTFL	13.40 (3.47)	88.39 (4..67)	7068	43.84 (5.98)	51.04 (7.40)	18664	27.51(12.25)	79.18 (12.87)	6000
TRML	16.21 (3.77)	86.02 (5.27)	7068	44.26 (6.05)	50.50 (7.36)	18664	26.58(11.82)	79.89 (12.49)	6000
OMTL	13.18 (7.65)	90.24 (4.75)	7068	38.13 (5.03)	54.73 (4.28)	18664	18.61 (7.29)	84.45 (8.64)	6000
COL	5.94 (1.67)	94.93 (1.93)	7068	41.46 (4.13)	50.89 (3.00)	18664	18.43 (6.75)	84.14 (8.93)	6000
OSMTL-e	5.22 (2.06)	95.67 (1.78)	7068	38.18 (4.95)	55.03 (4.09)	18664	19.24 (7.15)	83.18 (9.04)	6000
CWMT	4.90 (1.94)	95.93 (1.69)	7068	**36.72** (4.87)	**55.87** (3.09)	18664	17.98 (6.57)	84.73 (8.39)	6000
ALP	7.49 (0.34)	93.32 (2.30)	5530.8 (53.18)	42.64 (3.39)	50.05 (4.02)	13684.1 (86.19)	21.64 (6.98)	81.77 (8.96)	3905.3 (46.40)
ACWMT	**4.89** (1.95)	**95.94** (1.67)	**3983.2 (31.23)**	36.73 (3.66)	55.48 (4.14)	**11388.5 (69.26)**	**17.97** (6.54)	**84.78** (8.23)	**3171.5 (29.09)**

Figure 1: Cumulative error rate, F1-measure and queried number along the entire online learning process



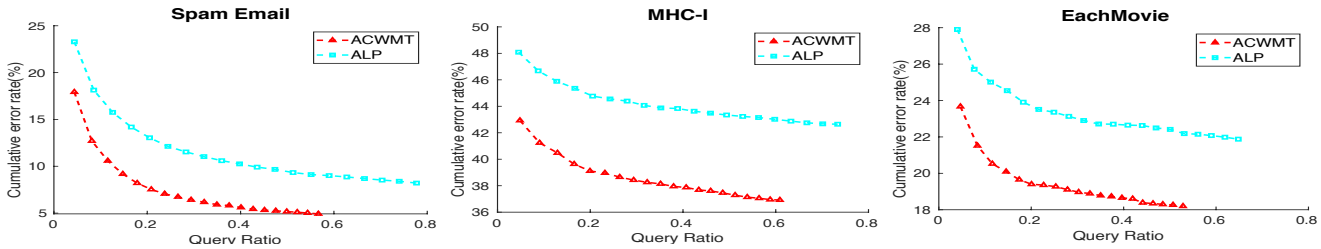
Spam Email[2], maintained by Internet Content Filtering Group, collects 7068 emails from mailboxes of 4 users (i.e., 4 tasks). Each mail entry is represented by a word document vector via the TF-IDF conversion technique. For each user, a model is proposed to classify each incoming email into two categories: *legitimate* or *spam*. Due to no time stamp on the emails, the email order is shuffled into a random sequence.
MHC-I[3], a biomarker dataset, contains 18664 peptide sequences from 12 MHC-I molecules (i.e., 12 tasks). Each peptide sequence is converted to a 400 dimensional feature vector (Li et al. 2011). We aim to classify whether a peptide sequence is *binder* or *non-binder* for each MHC-I molecule.
EachMovie[4] is a movie recommendation dataset. We randomly prioritize 6000 user-rating pairs where 30 users rate exactly 200 movies each. The six possible ratings (i.e. $[1, 6]$) are converted into two classes, *like* or *dislike*, based on the rating order. For each user (as a task), we randomly select 1783 users who viewed the same 200 movies and use their ratings as the features of movie instances.

Evaluation Metrics

We evaluate the performance of the algorithms with three measurements: 1) cumulative error rate, reflecting the prediction accuracy of an online algorithm; 2) number of queried labels, reflecting the label-efficiency of an algorithm; and 3) F1-measure, the harmonic mean of precision and recall, suitable to evaluate the performance on class-imbalance data. For error rate and queried number, a small value indicates better performance of a method. For F1-

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

Figure 2: A comparison between ACWMT and ALP with respect to different queried ratios



measure, a higher value indicates a better result. In order to compare these algorithms fairly, we randomly shuffle the ordering of samples for each dataset. We repeat each experiment 10 times and calculate the average results.

Baseline and Parameter Setting

Our baselines cover three categories of MTL techniques: 1) two batch learning techniques: multitask feature learning (*MTFL*) (Argyriou, Evgeniou, and Pontil 2006) and trace-norm regularized MTL (*TRML*) (Zhou, Chen, and Ye 2011); 2) three online learning algorithms: online MTL (*OMTL*) (Saha et al. 2011), collaborative online MTL (*COL)* (Li et al. 2014), and online smoothed MTL (*OSMTL-e*) (Murugesan et al. 2016); and 3) one active online learning method: active learning from peers (ALP) (Murugesan and Carbonell 2017). To handle the online setting, we modify the batch learning setting of MTFL and TRML by periodically retraining online data after observing 100 samples. All parameters of the baselines are tuned according to their recommended instructions. CWMT and ACWMT are the proposed multitask algorithms in the fully-supervised setting and the partially-labeled setting, respectively. For simplicity, we set $\epsilon = \lambda = 100$ to avoid overfitting. In the query method, we set $h = 0.1$ for MHC-I and EachMovie, $h = 1$ for Spam-Email.

Comparison Result

The experimental results are presented in Table 2. We also show the evaluation measures with respect to the round of online learning in Fig. 1. The improvement of our algorithms over ALP is always significant over all datasets. This is consistent with previous observations in online multitask learning: the second-order algorithms are generally better than the first-order ones (Yang et al. 2015; Yang, Zhao, and Gao 2018). The reason is that the covariance matrix that encodes the confidence of parameters can guide the direction and magnitude of the parameter learning.

ACWMT always achieves a lower error rate with a smaller number of queries. The possible reasons are three folds. 1) The prediction variance is reduced by exploiting the Gaussian distribution of parameter weights. 2) The labeling cost is saved by the proposed adaptive-margin query. And 3) the aggressive updating strategy accelerates the learning progress. These techniques speed up the convergence of this method, so that the error rate and queried number can be reduced further when the model learns sufficient knowledge of data. It demonstrates both the computational efficiency and label efficiency of these algorithms.

We observe that ACWMT achieves comparable accuracy to CWMT with a small number of queries. The reason may be due to the class-imbalance issue, where the training instances from the minority class are much fewer than that from the majority class. ACWMT can learn aggressively on the minority class, while CWMT accesses all labels and the majority class may dominate the predictive model, leading to poor performance (Zhang, Yang, and Srinivasan 2016).

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

Algorithm	Spam Email	MHC-I	EachMovie
TRML	73.55	361.42	391.22
MTFL	78.01	198.90	302.17
COL	1.86	6.35	31.01
OSMTL-e	1.36	4.33	11.43
ACWMT	11.49	11.45	17.92

Sensitivity Analysis on Query Ratio

We study the impact of the parameter h. The algorithm with a lower value of h will conduct a small number of queries. Specifically, we set h to $\{10^{-4}, 10^{-3}, \ldots, 1\}$ and calculate the averaged query ratio over 10 times of random shuffles. The comparison result is shown in Fig. 2. We observe that ACWMT achieves better accuracy consistently under different ratios of queries. This is expected since ACWMT determines a query by leveraging the second-order information of the prediction, which is more effective to capture the informative instances than ALP that adopts first-order query strategy.

Time Complexity

We study the time complexity of the proposed algorithm in Table 3. We observe that the online methods run faster than the two batch learning algorithms. This is obvious since online models only learn on the current instance, while batch models learn with a substantial amount of samples. In addition, we observe that ACWMT is relatively slower than OSMTL-e. This is expected since ACWMT has to update the covariance matrix of parameter weights. However, the extra computational cost is worth it since the significant improvement has been achieved by considering the parameter confidence.

Conclusion

This work presents an online active learning method in the multitask setting, where the learner performs local task jointly with learning global knowledge of peer tasks. The intuition in this paper is that learning efficacy can be improved by accessing the knowledge of peer tasks. Our query strategy is benefited from two aspects: 1) query decision is made by consulting the local-global knowledge of multiple tasks; and 2) query confidence depends on both the margin and the variance of its prediction. Theoretical results show that our method that runs on a fraction of informative labels achieves a lower error bound than the fully-supervised counterparts. Finally, the promising empirical results validate the effectiveness of our methods on several real-world datasets.

Acknowledgements: The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST), under award number URF/1/3007-01-01.

References

Abernethy, J.; Bartlett, P.; and Rakhlin, A. 2007. Multitask learning with expert advice. In *COLT*, 484–498. Springer.

Agarwal, A.; Rakhlin, A.; and Bartlett, P. 2008. Matrix regularization techniques for online multitask learning. Technical report, EECS Department, University of California, Berkeley.

Argyriou, A.; Evgeniou, T.; and Pontil, M. 2006. Multi-task feature learning. In *NIPS*, 41–48.

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

Cesa-Bianchi, N.; Gentile, C.; and Zaniboni, L. 2006. Worst-case analysis of selective sampling for linear classification. *JMLR* 7(Jul):1205–1230.

Crammer, K., and Mansour, Y. 2012. Learning multiple tasks using shared hypotheses. In *Advances in Neural Information Processing Systems*, 1475–1483.

Crammer, K.; Dredze, M.; and Pereira, F. 2009. Exact convex confidence-weighted learning. In *NIPS*, 345–352.

Dekel, O.; Gentile, C.; and Sridharan, K. 2012. Selective sampling and active learning from single and multiple teachers. *JMLR* 13(Sep):2655–2697.

Dekel, O.; Long, P. M.; and Singer, Y. 2007. Online learning of multiple tasks with a shared loss. *JMLR* 8:2233–2264.

Ding, L.-Z.; Liao, S.; Liu, Y.; Yang, P.; and Gao, X. 2018. Randomized kernel selection with spectra of multilevel circulant matrices. In *AAAI*.

Evgeniou, T., and Pontil, M. 2004. Regularized multi–task learning. In *SIGKDD*, 109–117. ACM.

Hoi, S. C.; Sahoo, D.; Lu, J.; and Zhao, P. 2018. Online learning: A comprehensive survey. *arXiv preprint arXiv:1802.02871*.

Li, G.; Chang, K.; Hoi, S. C. H.; Liu, W.; and Jain, R. 2011. Collaborative online learning of user generated content. In *CIKM*, 285–290.

Li, G.; Hoi, S. C.; Chang, K.; Liu, W.; and Jain, R. 2014. Collaborative online multitask learning. *TKDE* 26(8):1866–1876.

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

Murugesan, K., and Carbonell, J. 2017. Active learning from peers. In *Advances in Neural Information Processing Systems*, 7011–7020.

Murugesan, K.; Liu, H.; Carbonell, J.; and Yang, Y. 2016. Adaptive smoothed online multi-task learning. In *Advances in Neural Information Processing Systems*, 4296–4304.

Saha, A.; Rai, P.; Daumé III, H.; and Venkatasubramanian, S. 2011. Online learning of multiple tasks and their relationships. In *AISTATS*.

Sherman, J., and Morrison, W. J. 1950. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics* 21(1):124–127.

Tseng, P. 2001. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications* 109(3):475–494.

Yang, P., and Zhao, P. 2015. A min-max optimization framework for online graph classification. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 643–652. ACM.

Yang, P.; Zhao, P.; Zheng, V. W.; and Li, X.-L. 2015. An aggressive graph-based selective sampling algorithm for classification. In *Data Mining (ICDM), 2015 IEEE International Conference on*, 509–518. IEEE.

Yang, P.; Zhao, P.; and Gao, X. 2017. Robust online multitask learning with correlative and personalized structures. *IEEE Transactions on Knowledge and Data Engineering* 29(11):2510–2521.

Yang, P.; Zhao, P.; and Gao, X. 2018. Bandit online learning on graphs via adaptive optimization. International Joint Conferences on Artificial Intelligence Organization.

Zhang, Y., and Yeung, D.-Y. 2010. A convex formulation for learning task relationships in multi-task learning. In *UAI*, 733–742. AUAI Press.

Zhang, X.; Yang, T.; and Srinivasan, P. 2016. Online asymmetric active learning with imbalanced data. In *SIGKDD*, 2055–2064. ACM.

Zhao, P.; Hoi, S. C.; Jin, R.; and Yang, T. 2011. Online auc maximization. In *ICML*, 233–240. Omnipress.

Zhou, J.; Chen, J.; and Ye, J. 2011. *MALSAR: Multi-tAsk Learning via StructurAl Regularization*. Arizona State University.