8-2014

Collaborative Online Multitask Learning

Guangxia LI
Nanyang Technological University

Steven C. H. HOI
Singapore Management University, CHHOI@smu.edu.sg

Kuiyu CHANG
Nanyang Technological University

Wenting LIU
Nanyang Technological University

Ramesh JAIN
University of California, Irvine

DOI: https://doi.org/10.1109/TKDE.2013.139

Published in IEEE Transactions on Knowledge and Data Engineering, Volume 26, Issue 8, August 2014, Pages 1866-1876
https://doi.org/10.1109/TKDE.2013.139

Collaborative Online Multitask Learning

Guangxia Li, Steven C.H. Hoi, Kuiyu Chang, Wenting Liu, and Ramesh Jain

Abstract—We study the problem of online multitask learning for solving multiple related classification tasks in parallel, aiming at classifying every sequence of data received by each task accurately and efficiently. One practical example of online multitask learning is the micro-blog sentiment detection on a group of users, which classifies micro-blog posts generated by each user into emotional or non-emotional categories. This particular online learning task is challenging for a number of reasons. First of all, to meet the critical requirements of online applications, a highly efficient and scalable classification solution that can make immediate predictions with low learning cost is needed. This requirement leaves conventional batch learning algorithms out of consideration. Second, classical classification methods, be it batch or online, often encounter a dilemma when applied to a group of tasks, i.e., on one hand, a single classification model trained on the entire collection of data from all tasks may fail to capture characteristics of individual task; on the other hand, a model trained independently on individual tasks may suffer from insufficient training data. To overcome these challenges, in this paper, we propose a collaborative online multitask learning method, which learns a global model over the entire data of all tasks. At the same time, individual models for multiple related tasks are jointly inferred by leveraging the global model through a collaborative online learning approach. We illustrate the efficacy of the proposed technique on a synthetic dataset. We also evaluate it on three real-life problems—spam email filtering, bioinformatics data classification, and micro-blog sentiment detection. Experimental results show that our method is effective and scalable at the online classification of multiple related tasks.

Index Terms—Artificial intelligence, learning systems, online learning, multitask learning, classification

1 Introduction

Classical machine learning methods are often formulated as a single task learning problem, which by definition learns one task at a time. On the contrary, multitask learning aims to solve multiple related learning tasks in parallel. Many real-world problems are essentially multitask learning, although they are often broken into smaller single learning tasks, which are then solved individually by classical learning methods. Multitask learning has been extensively studied in machine learning and data mining over the past decade [1]–[4]. Empirical findings have demonstrated the advantages of multitask learning over single task learning across a variety of application domains.

The classical multitask learning methodology [1] often makes two assumptions. First, it assumes there is one primary task and other related tasks are simply secondary ones whose training data are exploited by multitask learning to improve the primary task. Thus, the classical multitask learning approach focuses on learning the primary task without caring how the other tasks are learned. Second, the classical multitask learning problem is often studied in a batch learning setting, which assumes that the training data of all tasks are available. On one hand, this assumption is not realistic for many real-world problems where data arrives sequentially. On the other hand, the batch multitask learning algorithms usually have fairly intensive training cost and poor scalability performance, as far as large real applications are concerned.

In this paper, we investigate the problem of online multitask learning, which differs from the classical multitask learning in two aspects. First, our goal is to improve the learning performance of all tasks instead of focusing on a single primary task. Second, we frame the multitask learning problem in an online learning setting by assuming that the data for each task arrives sequentially, which is a more realistic scenario for real-world applications. Unlike batch learning techniques, online learning methods learn over a sequence of data by processing each sample upon arrival. At each round, the learner first receives one instance, makes a prediction, and receives the true label. The error information is then used to update the learning model.

Our early study on this work [5], [6] was first motivated by the need to classify online user-generated content (UGC), e.g., micro-blog posts or spam email tags. For UGC, each individual exhibits uniqueness, but also shares certain characteristics with others in the group. It is thus desirable to develop an efficient and scalable classifier that can solve individual task by adapting to the global knowledge shared by all users. Consider the real problem of micro-blog sentiment analysis on a group of users where the goal is to classify micro-blog posts generated by each user into several emotional or non-emotional categories in a near real-time manner. When solving this problem by classical machine learning techniques, we will face a dilemma, i.e., a single global classification model trained on the entire

- *G. Li, K. Chang and W. Liu are with the School of Computer Engineering, Nanyang Technological University, Singapore, 639798. E-mail: {ligu0005, askychang, wliu7}@ntu.edu.sg.*
- *S. C. H. Hoi is with the School of Information Systems, Singapore Management University, Singapore, 178902. E-mail: stevenhoi@gmail.com.*
- *R. Jain is with the School of Information and Computer Sciences, University of California, Irvine, CA 92697, USA. E-mail: jain@ics.uci.edu.*

collection of data from all users may fail to capture the peculiarity of individual users and thus often works poorly. On the other hand, a fully personalized model for each user may be inaccurate due to insufficient training data, especially at the early stage of the learning task. This thus motivates us to study online multitask learning techniques.

We propose a novel collaborative online multitask learning (COML) technique to attack the aforementioned challenges. The basic idea is to first build a generic *global* model from large amount of data gathered from all users, and then subsequently leverage the global model to build the *personalized* classification models for individual users through a *collaborative* learning process. We formulate this idea into an optimization problem under an online learning setting, and propose two different COML algorithms by exploring different kinds of online learning methodologies.

To evaluate the efficacy of the proposed technique, we conduct experiments by comparing our algorithms against a variety of state-of-the-art techniques on a synthetic dataset and three real-life applications, including online spam email filtering, peptide binding prediction in bioinformatics, and micro-blog sentiment detection. Our results show that the proposed COML algorithms outperform (1) a single task online learning approach that simply learns a global model over the entire collection of data gathered from all the tasks, (2) a single task online learning approach that solves each task independently, and (3) a state-of-the-art online multitask learning approach.

The rest of this paper is organized as follows. Section 2 introduces related work. Section 3 presents the proposed algorithms. Section 4 gives experimental results and discussions. Section 5 concludes this paper.

2 Related Work

Our work is closely related to two groups of research in machine learning and data mining, i.e., (1) online learning, and (2) multitask learning. We briefly survey the representative work in each area.

Online learning has been extensively studied [7]–[14]. Unlike batch learning methods, which assume all training samples to be available before the learning task begins, online learning algorithms incrementally build a model from a stream of samples, allowing for simple and fast model updates. They are thus naturally capable of dealing with large datasets and problems whose data arrive sequentially. The origin of online learning can date back to the well-known Perceptron algorithm [10], which updates the model weight by moving it a tad closer to each misclassified sample. Descendant Perceptron-like methods employ more sophisticated update strategies. For example, a variety of online learning algorithms have been proposed based on the maximum margin learning principle that has been successfully applied to batch mode learning [8], [9], [13], [14]. Specifically, the Relaxed Online Maximum Margin (ROMMA) algorithm [14] repeatedly chooses a hyper-plane that correctly classifies existing training examples with the maximum margin. The family of Passive-Aggressive (PA) algorithms [8] maintains a trade-off between the amount of progress made on each training round and information retained from previous rounds. In particular, the PA algorithm updates the model whenever a new example is misclassified or when its classification score is smaller than some predefined margin. Empirical studies showed that the maximum margin based online learning algorithms are generally more effective than the Perceptron algorithm.

The above online learning algorithms in general belong to the family of first-order online learning techniques. Recent years have witnessed emerging studies on exploring second-order information for online learning. The second-order online learning algorithms that improve upon the Perceptron-like methods include the second-order Perceptron (SOP) [15], confidence weighted (CW) learning [16] and its successors [17]–[20]. The confidence weighted learning algorithm maintains a probabilistic measure of confidence in each component of its weight vector using a Gaussian distribution. The weight distribution is updated by minimizing the Kullback-Leibler divergence between the new weight distribution and the old one under the constraint that the probability of correct classification is greater than a threshold. AROW [19], which stands for "adaptive regularization of weight vectors", softens the hard constraint in confidence weighted learning as regularizers. It also uses an improved update strategy, leading to extra robustness in the case of non-separable data. Empirically, AROW has been demonstrated to show state-of-the-art performance for typical online learning tasks.

The problem of jointly solving several related learning tasks by leveraging the commonality among tasks has been studied in the machine learning community under the guise of multitask learning [2], [21]–[27]. The relationship of the tasks has been modeled in a number of ways. The pioneering work [1] assumed there is one primary task and other secondary tasks, which are solely used to improve the learning of the primary task. Evgeniou *et al*. [2] introduced multitask kernel and considered batch multitask learning as a regularized optimization problem. Ando *et al*. [26] formulated multitask relations by enforcing predictive functions for different tasks to belong to the same hypothesis set. Kang *et al*. [28] studied the problem of multitask learning of shared feature representations among tasks, while simultaneously determining "with whom" each task should share. Some other studies tried to explore underlying spectral dependencies among tasks [21], [29], [30]. In [31], the authors used feature hashing to solve multitask learning problem. For each task, they minimized the interaction between its parameter vector and the combination of other tasks' parameter vectors.

Unlike the existing batch multitask learning studies, our work is closer to the online multitask learning methodology. The online multitask learning problem was first addressed in [32], who assumed a very general setting wherein the tasks are related by a global loss function and the goal is to reduce the cumulative loss (for all tasks involved) over all rounds of the online algorithm. Following the same line of thought, the studies in [33], [34] formulated the multitask learning problems as online learning with expert advice. Regret bounds are given under the assumption that there is a set of best experts who perform well on the entire set of tasks. Saha *et al*. [35] proposed to learn the task models as well as the task relatedness in a coherent way. Mistake bounds for online multitask learning have

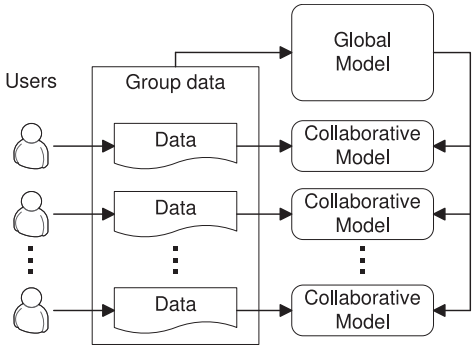


Fig. 1. Leveraging data from a series of users via collaborative online multitask learning.

also been investigated in [34], [36]. Instead of explicitly modeling the specific task relatedness measure, our technique jointly learns a generic global model shared by many parallel learners and individual collaborative models by reinforcing each learning task through a collaborative learning process. Our approach not only enjoys a considerable improvement in classification performance, but also retains the hallmark low computational cost of online learning algorithms.

3 Methodology

3.1 Overview

The motivation of our solution is two-fold. First, as tasks often exhibit varying patterns, it is neither practical nor effective to learn a single global model for classification. Second, it is also not always possible to learn a good classification model for each task since training data available for a single task are often limited. For such case, it is reasonable to pool together data across many related tasks. Hence, a better solution is to combine these two approaches coherently.

Fig. 1 illustrates the idea of our method. Specifically, the collaborative online multitask learning operates in a sequential manner. At each learning round, it collects the current global set of data; one from each of the engaged users/tasks, which are employed to update the global classification model. At the same time, a collaborative personalized model is maintained for each user/task. The individual collaborative classification model is subsequently updated using the latest individual data and the global model parameters. Therefore, our approach can leverage global knowledge for classification, while adapting to individual nuances via the collaborative learning way.

3.2 Formulation and Algorithms

We now formulate the problem in a binary classification setting. Our algorithm can be easily extended to address the multiclass problems by adopting techniques described in [8], [18].

Online multitask classification proceeds in rounds by observing a sequence of examples, each belonging to some user/task from a set of K users/tasks. On each round, there are K separate online binary classification problems being solved jointly. We assume that data from all users/tasks can be represented in the same global feature space, so that it is possible to use the shared information between tasks to enhance each learning task. Denote by $(\mathbf{x}_t^k, y_t^k)$ a training instance belonging to the k-th user at round t, where $\mathbf{x}_t^k \in \mathbb{R}^d$ is a d-dimensional vector representing the example and $y_t^k \in \{1, -1\}$ refers to its class label. We henceforth omit the superscript of $\mathbf{x}_t^k$ below for brevity.

Our goal is to learn a set of classification models to maximize the online prediction accuracy of every task, i.e., $f^{(k)}(\cdot):\mathbb{R}^d \mapsto \{1, -1\}, k = 1, \ldots, K$. In this work, we consider a linear classification model for each task, which is parameterized by a weight vector $\mathbf{w}$, i.e., $f(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x})$.

3.2.1 Building a Global Model by Online Learning

The first step of the collaborative online multitask learning builds a global classification model to exploit the commonality among tasks. We adopt the online passive aggressive (PA) framework [8] to build a global model using data collected from *all* users at round t, that is

$$f_t(\mathbf{x}) = \text{sign}(\mathbf{u}_t \cdot \mathbf{x})$$

where $\mathbf{u}_t \in \mathbb{R}^d$ is the weight vector of the global model learned at round t.

Specifically, at round t, the algorithm uses the latest training instance $(\mathbf{x}_t, y_t)$ to update the classification model as follows

$$\mathbf{u}_{t+1} = \underset{\mathbf{u}\in\mathbb{R}^d}{\text{argmin}} \quad \frac{1}{2}\|\mathbf{u} - \mathbf{u}_t\|^2 + C\xi \tag{1a}$$

$$\text{s.t.} \quad \ell(\mathbf{u}; (\mathbf{x}_t, y_t)) \leq \xi \tag{1b}$$

$$\xi \geq 0 \tag{1c}$$

where C is a positive parameter controlling the influence of the slack variable ξ on the objective function, and ℓ is the hinge loss function defined as

$$\ell(\mathbf{u}; (\mathbf{x}_t, y_t)) = \max(0, 1 - y_t \mathbf{u} \cdot \mathbf{x}_t)$$

The above formulation aims to achieve two objectives: (1) variation of the new weight vector $\mathbf{u}_{t+1}$ from the previous weight vector $\mathbf{u}_t$ should be as small as possible, and (2) the new weight vector should correctly classify the current example $\mathbf{x}_t$ with a sufficiently large margin. By doing so, it maintains a trade-off between the amount of progress made on each round and information retained from previous rounds. The closed-form solution of the optimization problem (1) is

$$\mathbf{u}_{t+1} = \mathbf{u}_t + \tau_t y_t \mathbf{x}_t$$

where τ_t is given by $\tau_t = \min\left\{C, \frac{\ell_t}{\|\mathbf{x}_t\|^2}\right\}$. The proof can be found in [8].

3.2.2 Learning the Collaborative Models

The critical step of our collaborative online multitask learning is to apply the existing global model to collaboratively learn the each of the K individual user models. Using the same PA formulation, the goal is to learn a classification model for the k-th user as

$$f_t^{(k)}(\mathbf{x}) = \text{sign}\left(\mathbf{w}_t^{(k)} \cdot \mathbf{x}\right)$$

where $\mathbf{w}_t^{(k)} \in \mathbb{R}^d$ is the weight vector of the k-th user's collaborative model learned at round t. For simplicity, we use $\mathbf{w}_t$ to denote $\mathbf{w}_t^{(k)}$ henceforth.

The next step is to use the shared information learned by the global model to enhance each individual learning model. We formulate the collaborative learning model as a convex optimization problem that minimizes the deviation of the new weight vector from the prior collaborative one and the global one, as follows

$$\mathbf{w}_{t+1} = \underset{\mathbf{w}\in\mathbb{R}^d}{\operatorname{argmin}} \quad \frac{\eta_1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2 + \frac{\eta_2}{2}\|\mathbf{w}-\mathbf{u}_t\|^2 + C\xi \tag{2a}$$

$$\text{s.t.} \quad \ell(\mathbf{w};(\mathbf{x}_t,y_t)) \le \xi \tag{2b}$$

$$\xi \ge 0 \tag{2c}$$

where η_1 and η_2 are two parameters that balance the trade-off between the global model $\mathbf{u}$ and the collaborative model $\mathbf{w}$, and parameter $C \ge 0$ controls the influence of the slack variable ξ on the objective function.

The above formulation aims to achieve a balance between the global and individual models, i.e., in spite of its uniqueness, each individual also shares some commonality with other members in the group. It coherently combines the collaborative model with the global one. In particular, if we set $\eta_2 = 0$, the optimization reduces to the approach of learning an individual classification model without engaging the global model; if we set $\eta_1 = 0$, it reduces to the global model. Accordingly, we can fine tune the contribution of each model by setting appropriate parameters.

Applying the Lagrangian multiplier technique, the update rule for optimization problem (2) can be derived as

$$\mathbf{w}_{t+1} = \frac{\eta_1\mathbf{w}_t + \eta_2\mathbf{u}_t + \tau y_t\mathbf{x}_t}{\eta_1+\eta_2} \tag{3}$$

where τ is given by

$$\tau = \min\left\{C, \frac{\eta_1+\eta_2 - y_t(\eta_1\mathbf{w}_t+\eta_2\mathbf{u}_t)\cdot\mathbf{x}_t}{\|\mathbf{x}_t\|^2}\right\}$$

The pseudo code for the proposed collaborative online multitask learning is given in Algorithm 1.

Algorithm 1: Collaborative online multitask learning

Input: training data of K users $(\mathbf{x}^k, y^k)$, $k = 1,\ldots,K$, parameters $\eta_1 \ge 0$, $\eta_2 \ge 0$, $C \ge 0$

Initialize $\mathbf{w}_0^k \leftarrow \mathbf{0}$, $\mathbf{u}_0 \leftarrow \mathbf{0}$;

for $t \leftarrow 1$ **to** T **do**

  // Update the collaborative models

  for $k \leftarrow 1$ **to** K **do**

    receive global model weight vector $\mathbf{u}_t$;

    receive training instance $(\mathbf{x}_t^k, y_t^k)$;

    set loss $\ell(\mathbf{w}^k;(\mathbf{x}_t^k,y_t^k)) = \max(0, 1 - y_t^k\mathbf{w}^k\cdot\mathbf{x}_t^k)$;

    if $\ell(\mathbf{w}^k;(\mathbf{x}_t^k,y_t^k)) = 0$ **then**

      $\tau = 0$

    else

      $\tau = \min\left\{C, \frac{\eta_1+\eta_2 - y_t(\eta_1\mathbf{w}_t+\eta_2\mathbf{u}_t)\cdot\mathbf{x}_t}{\|\mathbf{x}_t\|^2}\right\}$

    end

    update $\mathbf{w}_{t+1} = \frac{\eta_1\mathbf{w}_t+\eta_2\mathbf{u}_t+\tau y_t\mathbf{x}_t}{\eta_1+\eta_2}$;

  end

  // Update the global model

  for $k \leftarrow 1$ **to** K **do**

    receive training instance $(\mathbf{x}_t^k, y_t^k)$;

    set loss $\ell(\mathbf{u}_t;(\mathbf{x}_t^k,y_t^k)) = \max(0, 1 - y_t^k\mathbf{u}_t\cdot\mathbf{x}_t^k)$;

    if $\ell(\mathbf{u}_t;(\mathbf{x}_t^k,y_t^k)) > 0$ **then**

      $\sigma = \min\{C, \frac{1-y_t^k\mathbf{u}_t\cdot\mathbf{x}_t^k}{\|\mathbf{x}_t^k\|^2}\}$;

      update $\mathbf{u}_{t+1} = \mathbf{u}_t + \sigma y_t^k\mathbf{x}_t^k$;

    end

  end

end

3.3 Extending Confidence Weighted Learning

A current trend in online learning research is to use parameter confidence information to guide online learning process. Confidence weighted learning, proposed by Crammer *et al*. [16], [17], [19], models the linear classifier hypotheses uncertainty with a multivariate Gaussian distribution over weight vectors, which is then used to control the direction and scale of parameter updates. Conceptually, to classify an instance $\mathbf{x}$, a confidence weighted classifier draws a parameter vector $\mathbf{w} \sim N(\mu, \Sigma)$ and predicts the label according to sign($\mathbf{w}\cdot\mathbf{x}$). In practice, however, the average weight vector $E(\mathbf{w}) = \mu$ is used to make the prediction. Confidence weighted learning algorithms have been shown to perform well on many tasks. In this section, we extend the proposed collaborative online multitask learning with the confidence weighted hypothesis.

We make use of the state-of-the-art confidence weighted learning algorithm—Adaptive Regularization of Weights (AROW) [19] in our work. It solves the following unconstrained objective function on each round

$$\underset{\mu\in\mathbb{R}^d,\,\Sigma\in\mathbb{R}^{d\times d}}{\operatorname{argmin}} \quad \mathrm{D}_{\mathrm{KL}}(\mathcal{N}(\mu,\Sigma)\,\|\,\mathcal{N}(\mu_t,\Sigma_t)) + \frac{1}{2r}\ell^2(\mu;(\mathbf{x}_t,y_t)) + \frac{1}{2r}\mathbf{x}_t^{\mathrm{T}}\Sigma\mathbf{x}_t$$

where the first term ensures that the updated distribution is similar to the current distribution $\mathcal{N}(\mu_t, \Sigma_t)$ in the Kullback-Leibler (KL) divergence sense. The second term $\ell^2(\mu;(\mathbf{x}_t,y_t)) = (\max(0, 1 - y_t\mu\cdot\mathbf{x}_t))^2$ is the squared hinge loss suffered from using the weight vector μ to predict the output for input $\mathbf{x}_t$ when the true label is y_t. The third term is related to a probabilistic constraint used in confidence weighted learning, i.e., a classifier drawn from the updated distribution should classify the example correctly with a high probability (see [16], [19] for details).

As in collaborative online multitask learning, we maintain a global classification model parameterized by a Gaussian distribution over weight vectors with mean $\tilde{\mu}$ and covariance $\tilde{\Sigma}$, and use the global model information together with each user data to train the collaborative models. The weight distribution is updated by minimizing the KL divergence between the new and old weight distributions, together with the KL divergence between the weight distributions of individual and global models. On each round, the updated collaborative model parameters

Algorithm 2: Confidence-weighted collaborative online multitask learning

Input: training data of K users $(\mathbf{x}^k, y^k)$, parameters η_1, η_2, r
Initialize $\mu_0^k \leftarrow \mathbf{0}$, $\Sigma_0^k \leftarrow I$, $\tilde{\mu}_0 \leftarrow \mathbf{0}$, $\tilde{\Sigma}_0 \leftarrow I$;
for $t \leftarrow 1$ **to** T **do**
  // Update the collaborative models
  for $k \leftarrow 1$ **to** K **do**
    receive global model parameters $(\tilde{\mu}_t, \tilde{\Sigma}_t)$;
    receive training instance $(\mathbf{x}_t^k, y_t^k)$;
    compute margin $m_t^k = \mu_{t-1}^k \cdot \mathbf{x}_t^k$;
    if $m_t^k y_t^k < 1$ **then**
      $\mu_{t+1}^k = MN + \frac{1 - y_t^k MN\mathbf{x}_t^k}{r + \mathbf{x}_t^{k\mathrm{T}} M \mathbf{x}_t^k} N y_t^k \mathbf{x}_t^k$
      $\Sigma_{t+1}^k = (\eta_1 + \eta_2)(M - \frac{M\mathbf{x}_t^k \mathbf{x}_t^{k\mathrm{T}} M}{r + \mathbf{x}_t^{k\mathrm{T}} M \mathbf{x}_t^k})$
    end
  end
  // Update the global model
  for $k \leftarrow 1$ **to** K **do**
    receive training instance $(\mathbf{x}_t^k, y_t^k)$;
    compute margin $\tilde{m}_t = \tilde{\mu}_{t-1} \cdot \mathbf{x}_t$;
    if $\tilde{m}_t y_t < 1$ **then**
      $\tilde{\mu}_{t+1} = \tilde{\mu}_t + \frac{1 - y_t \mathbf{x}_t^{\mathrm{T}} \tilde{\mu}_t}{r + \mathbf{x}_t^{\mathrm{T}} \tilde{\Sigma}_t \mathbf{x}_t} \tilde{\Sigma}_t y_t \mathbf{x}_t$
      $\tilde{\Sigma}_{t+1} = \tilde{\Sigma}_t - \frac{\tilde{\Sigma}_t \mathbf{x}_t \mathbf{x}_t^{\mathrm{T}} \tilde{\Sigma}_t}{r + \mathbf{x}_t^{\mathrm{T}} \tilde{\Sigma}_t \mathbf{x}_t}$;
    end
  end
end

(μ_t, Σ_t) are set to the result of the following optimization problem

$$\begin{aligned}\underset{\mu \in \mathbb{R}^d, \Sigma \in \mathbb{R}^{d \times d}}{\operatorname{argmin}} \; & \eta_1 \mathrm{D}_{\mathrm{KL}}(\mathcal{N}(\mu, \Sigma) \parallel \mathcal{N}(\mu_t, \Sigma_t)) + \frac{1}{2r} \mathbf{x}_t^{\mathrm{T}} \Sigma \mathbf{x}_t \\ & + \eta_2 \mathrm{D}_{\mathrm{KL}}(\mathcal{N}(\mu, \Sigma) \parallel \mathcal{N}(\tilde{\mu}_t, \tilde{\Sigma}_t)) + \frac{1}{2r} \ell^2(\mu; (\mathbf{x}_t, y_t))\end{aligned}$$

where η_1, η_2 and r are tradeoff parameters.

Following Crammer *et al*. [19], we use a correspondingly conservative update for the confidence parameter Σ, updating it only when μ changes. Assuming $1 - y_t \mu \cdot \mathbf{x}_t > 0$, taking the derivative of the objective function with respect to μ and Σ, respectively, and setting them to zero, we have the following update rules

$$\mu_{t+1} = MN + \frac{1 - y_t MN\mathbf{x}_t}{r + \mathbf{x}_t^{\mathrm{T}} M \mathbf{x}_t} N y_t \mathbf{x}_t$$

$$\Sigma_{t+1} = (\eta_1 + \eta_2) \left(M - \frac{M\mathbf{x}_t \mathbf{x}_t^{\mathrm{T}} M}{r + \mathbf{x}_t^{\mathrm{T}} M \mathbf{x}_t} \right)$$

where M and N are given by

$$M = \left(\eta_1 \Sigma_t^{-1} + \eta_2 \tilde{\Sigma}_t^{-1} \right)^{-1}$$

$$N = \eta_1 \Sigma_t^{-1} \mu_t + \eta_2 \tilde{\Sigma}_t^{-1} \tilde{\mu}_t$$

The pseudo code for confidence weighted collaborative online multitask learning is listed in Algorithm 2.

It is clear that the proposed algorithm in general enjoys a linear time complexity with respect to the number of instances and dimensions, which is not different from typical online learners. In terms of space complexity, the algorithm needs to maintain a single global model $\mathbf{u} \in \mathbb{R}^d$ and a set of K collaborative models $\mathbf{w}^{(k)} \in \mathbb{R}^d$ in memory. Hence, the storage requirement is $\mathcal{O}(d)$. For confidence weighted collaborative online multitask learning, we can use diagonal matrices instead of full matrices for the covariance Σ to save both computation and space costs greatly. It is also feasible to exploit the sparsity of data via sparse matrix data structures to reduce the memory cost in practice. Therefore, our algorithm is both efficient and scalable for large-scale online learning tasks.

4 Experimental Results

We evaluate the performance of our algorithm on a synthetic dataset and three real-life datasets. We start by introducing our experimental setup, followed by discussions on the results.

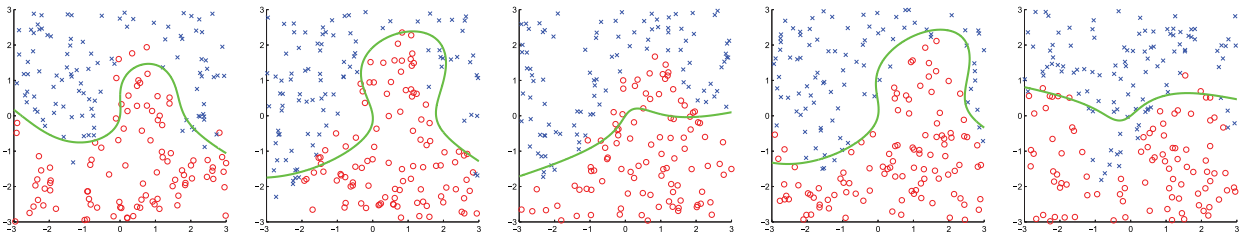
4.1 Benchmark Setup

We compare our COML algorithm with two batch learning methods (multitask feature learning [21], hereafter MTFL, and trace-norm regularized multitask learning [37], [38], hereafter TRML) and three online learning algorithms (online multitask learning [32], hereafter OML, PA [8], and AROW [19]). Due to the low computational speed, it is not feasible to update the two batch learning models repeatedly. We thus modify MTFL and TRML to handle online data by periodically retraining them after observing 100 samples.

To further examine the effectiveness of learning multiple related tasks together, we compare our method with a few variations of the PA and AROW algorithms as described below.

- **Global Model** It learns a single classification model from *all* tasks' data by applying the PA/AROW algorithm. At each online learning round, the algorithm receives a training sample from each task, and uses that sample to update its weight vector.
- **Personal Model** It employs the PA/AROW algorithm to train a personal classification model for *each* task only using its own data. In other words, every task is associated with a personalized classification model.
- **Simple Model** It simply switches between the Global and Personal models according to their cumulative error counts in previous online learning rounds. In particular, at each round, it sets its weight vector to that of the best model (Global or Personal), i.e., one with the least cumulative errors to-date. Benchmarking against this method is important as it will show whether the proposed COML algorithm is more effective than a naive combination.

We adopt the *cumulative error rate*, i.e., the ratio of the number of mistakes made by the online learning algorithm over total number of samples received to-date as a metric for comparing different algorithms. Despite its extensive usage in online learning studies, the cumulative error rate is not suitable for evaluating performances on class-imbalanced datasets. This is because for a highly imbalanced dataset, it is possible to deploy a trivial classifier



(a) (b) (c) (d) (e)

Fig. 2. Illustration of the synthetic dataset. Two classes are represented each by crosses and circles. Solid lines denote the COML classification boundaries after observing all training samples: (a) Task 1. (b) Task 2. (c) Task 3. (d) Task 4. (e) Task 5.

(i.e., blanket prediction of the majority class) that has low error rates but actually is of little use. We thus also report the F1-measure, which is the harmonic mean of precision and recall. It is typically harder for a classifier to achieve a good F1-measure compared to the error rate on an imbalanced dataset. The parameters C and r are set to 1 for all PA/AROW variations and our algorithm. Apart from testing various values of the parameters η_1 and η_2 for the two COML methods, all other parameters were set to default values. All experiment were conducted over 10 random permutations of the original dataset.

4.2 Synthetic Dataset

We used a synthetic dataset designed by Sheldon [25] to show that solving multiple related tasks jointly outperforms the solution that treats each task in isolation. The goal is to discriminate two classes (positive class and negative class) in a two-dimensional plane with non-linear decision boundaries as shown in Fig. 2. Denote by $\mathbf{x} = (x_1, x_2)$ a point in the two-dimensional space. The basic classification boundaries are generated according to the rule $g(\mathbf{x}; \mathbf{a}) = \text{sign}(x_2 - h(x_1; \mathbf{a}))$, where $h(x; \mathbf{a})$ is a simple family of nonlinear functions consisting of the first few terms of an arbitrary Fourier series defined as $h(x; \mathbf{a}) = a_1 \sin(x-a_0)+a_2 \sin(2(x-a_0))+a_3 \cos(x-a_0)+a_4 \cos(2(x-a_0))$. Rotation is also applied to the decision boundaries. Let R_θ be the operator that rotates a vector by θ radians in a counterclockwise direction about the origin. The final family of classifier is $f(\mathbf{x}; \mathbf{a}, \theta) = g(R_\theta \mathbf{x}; \mathbf{a})$ with θ as an additional parameter.

A total of five related tasks are involved in the experiment. Task parameters are generated via a random walk in parameter space with Gaussian increments [25]. Initial values are $\mathbf{a}^{(1)} = (0, 1, 1, 1, 1)$ and $\theta^{(1)} = 0$. For $t = 2, \ldots, 5$, $\mathbf{a}^{(t)} = \mathbf{a}^{(t-1)} + \epsilon_t, \epsilon_t \sim N(0, \sigma^2 I)$; $\theta^{(t)} = \theta^{(t-1)} + \delta_t, \delta_t \sim N(0, \sigma^2(\pi/4)^2)$. The parameter σ^2 controls step sizes and hence task similarity, and is set to 0.5. A training sample of size 200 is generated for each task by choosing 200 inputs $\mathbf{x}$ uniformly at random from the square $x_1, x_2 \in [-3, 3]$, and then labeling them according to $f(\mathbf{x}; \mathbf{a}, \theta)$. To fit the online learning scenario, generated samples are supplied to learning algorithms one by one. Since the problem is not linearly-separable, we added seven additional features/dimensions, which are derived from the original x_1 and x_2 via a mapping $(x_1, x_2) \mapsto (x_1, x_2, x_1x_2, x_1^2, x_2^2, x_1^3, x_2^3, x_1x_2^2, x_1^2x_2)$.

The average cumulative error rate, and its standard deviation across the 10 random permutations of the synthetic dataset are shown in Table 1. Fig. 3 depicts the detailed evolution of cumulative error rate along the entire online learning process of one trial. From these results, we can see that COML has the overall lowest averaged error rate of 13.81%, which is almost 5% better than the PA-Personal. To assess the statistical significance of the COML result, we performed an unpaired t-test at 5% significance level with PA-Personal as a baseline. The results of COML are proven to be statistically significant. When additional covariance information is incorporated, CW-COML is still better than the AROW based methods. In general, the proposed collaborative online multitask learning algorithms achieve lower cumulative error rates for all of the five tasks, which show that they are effective in learning problems with a common shared representation across multiple related tasks. The failure of the *Global* model is consistent with our intuition that learning related tasks via a single model is inappropriate as it ignores the individual task characteristic. A naive combination of global and individual models is also ineffective, as indicated by the suboptimal performance of the *Simple* method.

4.3 Spam Email Filtering

We apply online collaborative multitask learning to construct effective personalized spam email filters. The task is to classify each new incoming email message into two

TABLE 1
Cumulative Error Rate (%) and Its Standard Deviation (in Bracket) on the Synthetic Dataset

	Task1	Task2	Task3	Task4	Task5	Average
MTFL	14.40 (1.68)	18.65 (1.13)	14.50 (1.08)	9.30 (1.51)	16.50 (1.55)	14.67 (3.47)
TRML	15.35 (1.73)	19.15 (1.36)	14.10 (1.31)	9.75 (1.36)	16.80 (2.38)	15.03 (3.50)
OML	13.95 (1.94)	22.45 (1.95)	18.95 (1.98)	14.75 (1.18)	18.70 (2.77)	17.76 (3.46)
PA-Global	17.30 (2.25)	20.90 (2.66)	21.70 (2.56)	13.70 (2.04)	26.60 (1.91)	20.04 (4.85)
PA-Personal	14.10 (1.79)	22.60 (2.09)	19.45 (1.26)	14.50 (1.31)	19.35 (2.63)	18.00 (3.62)
PA-Simple	14.95 (1.79)	21.95 (2.79)	20.55 (1.79)	14.65 (2.66)	20.15 (2.66)	18.45 (3.40)
COML	9.45 (1.07)	17.75 (1.23)	12.45 (1.40)	9.20 (1.09)	20.20 (1.48)	13.81 (4.96)
AROW-Global	14.75 (2.55)	20.95 (2.27)	19.00 (1.29)	16.35 (2.70)	22.60 (1.33)	18.73 (3.22)
AROW-Personal	14.10 (1.05)	22.10 (2.73)	17.15 (1.58)	15.05 (1.21)	15.10 (1.56)	16.70 (3.22)
AROW-Simple	14.60 (2.09)	22.25 (2.47)	18.20 (1.57)	15.35 (1.58)	15.75 (1.81)	17.23 (3.11)
CW-COML	11.20 (1.74)	17.95 (2.40)	13.70 (1.01)	12.40 (1.81)	17.95 (2.30)	14.64 (3.15)

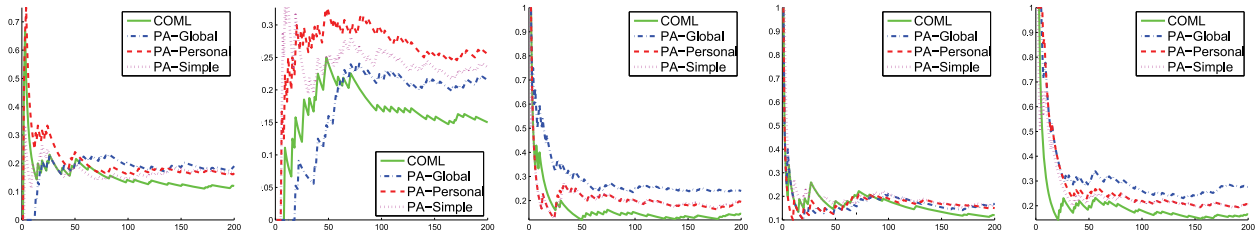


Fig. 3. Cumulative error rate (vertical axis) versus number of training samples observed (horizontal axis) on the synthetic dataset: (a) Task 1. (b) Task 2. (c) Task 3. (d) Task 4. (e) Task 5.

TABLE 2
Cumulative Error Rate and F1-Measure (%) on the Spam Email Dataset

	pu1			pu2			pu3			pua		
	Error	Legit F1	Spam F1	Error	Legit F1	Spam F1	Error	Legit F1	Spam F1	Error	Legit F1	Spam F1
MTFL	13.16 (1.21)	88.61 (1.06)	84.42 (1.42)	8.72 (0.58)	94.67 (0.38)	76.12 (1.27)	14.84 (0.91)	86.70 (0.88)	83.22 (0.94)	16.87 (0.78)	83.59 (0.77)	82.63 (0.83)
TRML	19.07 (0.58)	83.38 (0.60)	77.62 (0.54)	12.45 (1.28)	92.24 (0.87)	68.53 (2.18)	18.41 (1.35)	83.34 (1.28)	79.43 (1.43)	21.81 (1.17)	78.70 (1.12)	77.66 (1.29)
OML	5.00 (0.48)	95.55 (0.43)	94.29 (0.55)	8.01 (0.77)	95.07 (0.48)	78.52 (2.06)	3.55 (0.29)	96.84 (0.26)	95.96 (0.33)	8.24 (0.74)	91.72 (0.76)	91.81 (0.71)
PA-Global	6.15 (0.45)	94.54 (0.41)	92.98 (0.49)	8.59 (0.82)	94.72 (0.51)	76.98 (2.16)	4.12 (0.30)	96.33 (0.27)	95.30 (0.34)	9.75 (0.61)	90.20 (0.64)	90.29 (0.59)
PA-Personal	5.05 (0.49)	95.51 (0.44)	94.24 (0.56)	8.28 (0.85)	94.91 (0.52)	77.75 (2.33)	3.67 (0.36)	96.73 (0.32)	95.81 (0.41)	8.43 (0.86)	91.52 (0.89)	91.62 (0.84)
PA-Simple	5.17 (0.49)	95.41 (0.43)	94.10 (0.57)	8.30 (0.76)	94.90 (0.47)	77.80 (2.05)	3.67 (0.31)	96.73 (0.28)	95.81 (0.36)	8.61 (0.95)	91.34 (0.97)	91.43 (0.93)
COML	4.27 (0.51)	96.21 (0.45)	95.12 (0.59)	7.17 (0.62)	95.62 (0.38)	80.21 (1.75)	2.78 (0.17)	97.52 (0.15)	96.83 (0.19)	6.70 (0.64)	93.32 (0.64)	93.28 (0.64)
AROW-Global	5.31 (0.23)	95.26 (0.21)	93.96 (0.26)	6.82 (0.64)	95.81 (0.40)	81.82 (1.71)	3.88 (0.28)	96.52 (0.25)	95.60 (0.32)	8.23 (0.64)	91.88 (0.65)	91.67 (0.64)
AROW-Personal	4.50 (0.44)	95.99 (0.39)	94.87 (0.50)	7.03 (0.60)	95.67 (0.37)	81.27 (1.62)	2.99 (0.25)	97.32 (0.23)	96.62 (0.28)	6.94 (0.53)	93.15 (0.51)	92.96 (0.56)
AROW-Simple	4.60 (0.50)	95.90 (0.45)	94.77 (0.57)	7.07 (0.66)	95.65 (0.40)	81.07 (1.79)	3.02 (0.26)	97.30 (0.23)	96.58 (0.29)	7.08 (0.64)	93.02 (0.63)	92.83 (0.67)
CW-COML	5.10 (0.61)	95.45 (0.55)	94.19 (0.69)	6.58 (0.60)	95.91 (0.38)	83.25 (1.44)	4.14 (0.12)	96.29 (0.10)	95.31 (0.14)	7.95 (0.73)	92.08 (0.73)	92.01 (0.74)

categories: *legitimate* or *spam*. We use a dataset hosted by the Internet Content Filtering Group[1]. The dataset contains 7068 emails collected from mailboxes of four users (denoted by *pu1*, *pu2*, *pu3*, and *pua*). Strictly speaking, the set of all emails received by a user is not generated per se by that specific user. However, the characteristic of each user's email can be said to match his or her interest, whatever that may be. Each email entry is converted to a word document vector using the TF-IDF (term frequency-inverse document frequency) representation.

Since the email dataset has no time stamp, each user's email was shuffled into a random sequence. The cumulative error rate and F1-measure results of 10 shuffles are listed in Table 2. We also report each algorithm's run-time, i.e., the time consumed by both training and test phase during the complete online learning process in Table 3. From these results, we can make several observations.

First, the proposed COML consistently beats the other online learners in terms of error rate and F1-measure. In particular, in accordance to the results from the synthetic dataset, learning tasks collaboratively outperforms the baselines Global model, Personal model, and a simple combination of either.

Second, the performance of the proposed collaborative online multitask learning methods is better than that of the two batch learning algorithms (MTFL and TRML). It should be noted that compared to online learners who update models based only on the current sample, batch learning methods have the advantage of keeping a substantial amount of recent training samples, at the cost of storage space and higher complexity. In fact, the proposed COML algorithm is more efficient than batch incremental methods, e.g., it is more than 500 times faster than batch MTFL as shown in Table 3 (0.5 secs versus 271.84 secs). COML does not store recent training samples. It only uses the current training sample and a simple rule to update the model. In contrast, batch learning algorithms need to keep a certain number of recent training samples in memory, leading to extra burden on storage and complexity. What's more, both MTFL and TRML needs to solve an optimization problem in an iterative manner. For practical applications involving hundreds of millions of users and features, the batch learning algorithms are no longer feasible, while online learners remain highly efficient and scalable.

Table 3 shows that COML is slightly slower than the original PA algorithm. This is expected since COML has to update one additional global model. However, for a group of users, only one group model is needed. The extra computational cost is trivial compared to the combined cost to update every user model. Therefore, the proposed COML algorithm is efficient and applicable to solving large-scale problems.

1. http://labs-repos.iit.demokritos.gr/skel/i-config/

TABLE 3
Run-Time (in Seconds) for Each Algorithm

	COML	PA-G	PA-P	OML	MTFL	TRML
Spam	0.50	0.29	0.23	24.97	271.84	202.19
MHC-I	1.25	0.59	0.49	42.92	198.90	361.42
Twitter	18.61	8.10	8.68	50.01	8548.12	4804.25

TABLE 4
MHC-I Dataset Result Showing Average Cumulative Error Rate, F1-Measure (%) and their Standard Deviation (in Brackets) of 12 Tasks

	Error Rate	Positive Class F1-measure	Negative Class F1-measure
MTFL	43.84 (0.50)	51.04 (0.40)	59.12 (0.60)
TRML	44.26 (0.52)	50.50 (0.46)	58.80 (0.63)
OML	41.56 (0.20)	51.13 (0.28)	63.08 (0.25)
PA-Global	44.70 (0.40)	45.44 (0.34)	61.28 (0.49)
PA-Personal	41.62 (0.21)	51.08 (0.28)	63.02 (0.29)
PA-Simple	42.00 (0.27)	50.07 (0.29)	62.93 (0.30)
COML	37.61 (0.24)	55.46 (0.26)	66.83 (0.31)
AROW-Global	44.48 (0.38)	46.06 (0.45)	61.29 (0.40)
AROW-Personal	41.87 (0.36)	51.40 (0.34)	62.56 (0.42)
AROW-Simple	42.12 (0.42)	50.46 (0.41)	62.61 (0.46)
CW-COML	41.32 (0.37)	50.89 (0.49)	63.59 (0.40)

4.4 MHC-I Binding Prediction

Computational methods are widely used in bioinformatics to build models to infer properties from biological data. In this experiment, we evaluate several methods to predict peptide binding to human MHC (major histocompatibility complex) class I molecules. It is known that peptides binding to MHC-I molecules plays a crucial role in the vertebrate immune system [39]. The prediction of such binding has valuable application in vaccine designs, the diagnosis and treatment of cancer, etc. Recent work has demonstrated that there exists common information between related molecules (alleles) and such information can be leveraged to improve the peptide MHC-I binding prediction [4], [40].

We use a subset of the binary labeled MHC-I dataset[2]. The task is to predict whether a peptide binds to a certain human MHC-I molecule or not, i.e., *binder* or *non-binder*. The data consists of peptide sequences for 12 human MHC-I alleles. Each allele is indicated by a prefix letter followed by four digits. In total, there are 18664 samples (A0201=3793, A0202=1363, A0203=1363, A0206=1358, A0301=2633, A2402=763, A2902=535, A3002=415, A3101=2435, A3301=1179, A6801=1183, A6802=1644). On average, 33% of sample is labeled as binder (positive class). The peptide sequences are represented by strings of length 9 over the alphabet of 20 amino acids. To incorporate string features, we apply the bigram amino acid encoding to the protein sequence. The bigram features are a pair of values (v_i, c_i), where v_i is the i-th feature and c_i denotes the number of occurrences of this feature in the sequence for $i = 1, \ldots, 20^2$. For example, the protein sequence "*ALAKAAAAI*" has a bigram feature vector of $\{(AL, 1), (LA, 1), (AK, 1), (KA, 1), (AA, 3), (AI, 1)\}$.

We report the average cumulative error rate and F1-measure of 12 tasks in Table 4. To make a clear comparison between the proposed COML and PA baselines, we show the variation of their cumulative error rates along the entire online learning process averaged over the 10 runs in Fig. 4.

From these results, we first observe that the permutations of the dataset have little influence on the performance of each method, as indicated by the small standard deviation values in Table 4. The proposed COML outperformed all competitors in terms of error rate and F1-measures of both classes. Note that the majority of the dataset is negative class, thus predicting more examples as the majority class decreases the overall error rate, but also degrades the accuracy of the minority positive class. The confidence weighted online learning (AROW) actually performed slightly worse than the vanilla PA algorithms. However, among the four confidence weighted models, learning related tasks jointly still outperforms learning the tasks individually, as shown by the improvement of CW-COML model over the AROW-Global, AROW-Personal and AROW-Simple models.

TABLE 5
Twitter Sentiment Dataset Result Showing Average Cumulative Error Rate, F1-Measure (%) and their Standard Deviation (in Brackets) of 12 Tasks

	Error Rate	Positive Class F1-measure	Negative Class F1-measure
MTFL	33.63 (0.19)	38.03 (0.08)	76.06 (0.17)
TRML	32.95 (0.27)	38.23 (0.25)	76.68 (0.33)
OML	19.98 (0.25)	48.76 (0.70)	86.93 (0.17)
PA-Global	17.94 (0.21)	59.43 (0.50)	88.14 (0.14)
PA-Personal	19.65 (0.14)	53.85 (0.39)	86.96 (0.09)
PA-Simple	18.04 (0.24)	58.98 (0.63)	88.07 (0.16)
COML	16.37 (0.21)	61.12 (0.56)	89.25 (0.13)
AROW-Global	16.89 (0.31)	58.38 (1.04)	89.06 (0.18)
AROW-Personal	20.03 (0.44)	51.69 (0.87)	86.76 (0.32)
AROW-Simple	21.55 (0.64)	50.84 (0.80)	85.07 (0.48)
CW-COML	16.98 (0.35)	58.26 (1.15)	88.98 (0.20)

4.5 Micro-Blog Sentiment Detection

With the growing popularity of micro-blogs like *Twitter*[3] comes the demand to understand their users. We focus on the micro-blog sentiment detection problem, whose goal is to identify whether a users micro-blog post contains emotions or sentiments. This problem is challenging because a micro-blog post is often very short and each person may have his/her unique way of expressing sentiments. Moreover, the proportion of emotional posts is typically very small, and varies across individuals. Ideally, a personalized classifier should be created for each micro-blogger. However, there is a dearth of training data for each user, making the personalized sentiment model vastly inaccurate unless the model has been trained over hundreds of micro-blogs.

A post on *Twitter* is called a tweet. We crawled 32,567 tweets written by 12 influential users according to *wefollow.com*[4]. The latest tweet in our dataset was published in May, 2010, and the oldest one was in February, 2008. Each tweet was converted into a word vector using the TF-IDF representation. A human annotator labeled 7,628 (23.42%) tweets as emotional (positive class), while others

2. http://ailab.cs.iastate.edu/red/mhci.html

3. http://twitter.com

4. http://wefollow.com

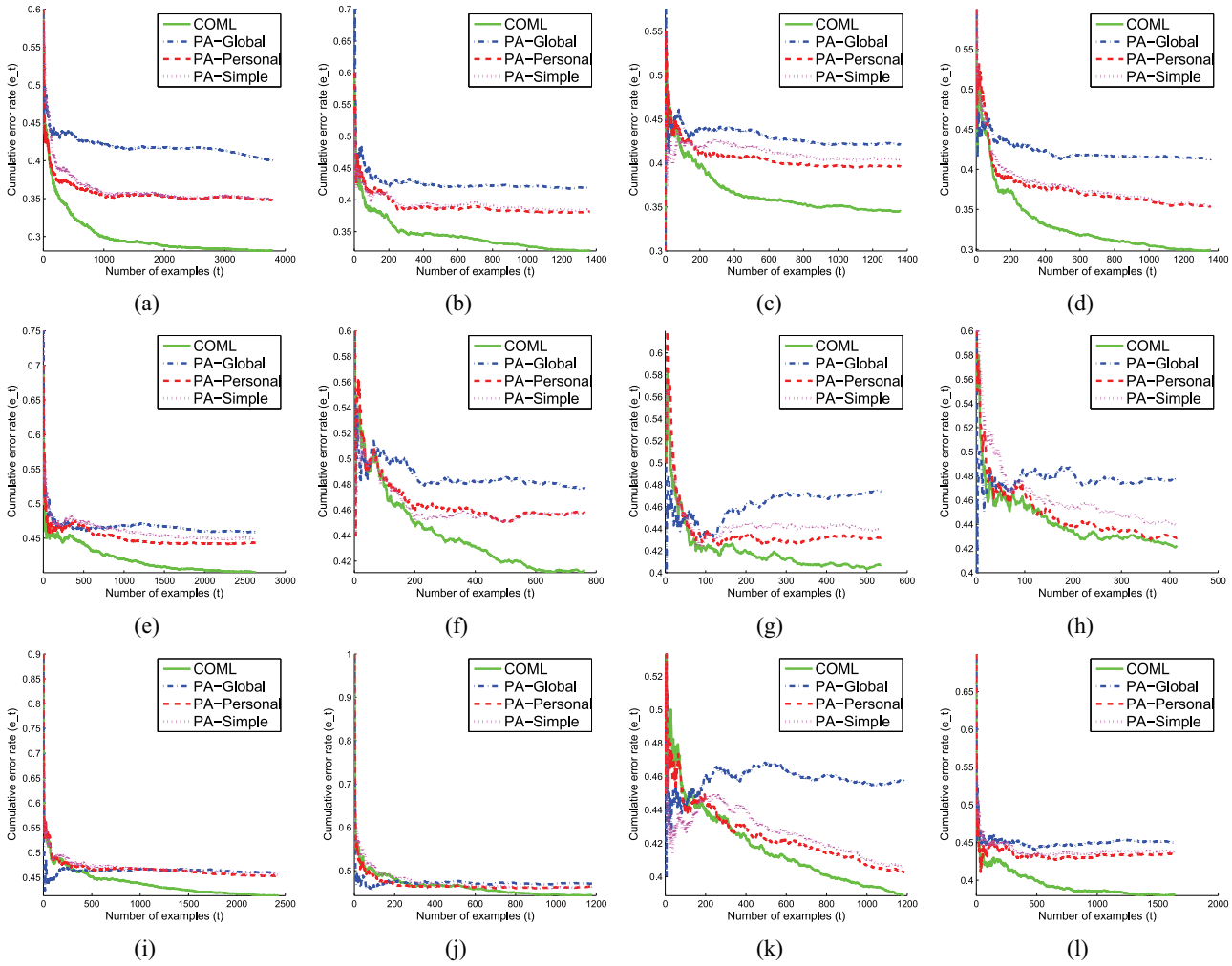


Fig. 4. Cumulative error rate on the MHC-I dataset along the entire online learning process: (a) A0201. (b) A0202. (c) A0203. (d) A0206. (e) A0301. (f) A2402. (g) A2902. (h) A3002. (i) A3101. (j) A3301. (k) A6801. (l) A6802.

as non-emotional (negative class). This result matches our conjecture that emotional tweets are generally a minority among *Twitter* users.

The average cumulative error rate, and its standard deviation across the 10 random permutations of the *Twitter* sentiment dataset are shown in Table 5. It can be seen that two of the models trained with PA and AROW—Global and Personal, match each other in strength, but loose out to the collaborative models. This result validates our previous concern that simply learning a single global model from all users' data is insufficient for solving the micro-blog sentiment detection task, since each user expresses emotions differently. The simple combination of Global model and Personal model—Simple model is able to approach the best model between Global and Personal, but fails to outperform the best one. This is because the "simple" scheme selects between the global and personal models by comparing their *historical* cumulative error counts. It is possible for a model with few mistakes in the earlier learning rounds to misclassify an incoming sample. Similarly, a model with large historical cumulative errors may score a lucky hit on that sample. This makes sticking to the first model an unwise choice.

5 Conclusion

In this paper, we proposed a collaborative online multitask learning method that is able to take advantage of individual and global models to achieve an overall improvement in classification performance for jointly learning multiple correlated tasks. We showed that it is able to outperform both the global and personal models by coherently integrating them in a unified collaborative learning framework. The experimental results demonstrate that our algorithms are both effective and efficient for three real-life applications, including online spam email filtering, MHC-I binding prediction, and micro-blog sentiment detection task.

Although the collaborative online multitask learning algorithm was firstly designed to solve the UGC classification problem, it has potential applications outside of the domains studied here. We hope to be able to extend our experiments to a more substantial size dataset and also to more applications. Our methods assume uniform relations across tasks. However, it is more reasonable to take into account the degree of relatedness among tasks. How to incorporate hierarchies and clusters of tasks is also worthy of further study. In conclusion, our collaborative online multitask learning method is a significant first

step towards a more effective online multitask classification approach.

ACKNOWLEDGMENTS

This research was supported by the Singapore Ministry of Education's Academic Research Fund Tier 2 Grant ARC 9/12 (MOE2011-T2-2-056). S. C. H. Hoi is the corresponding author.

REFERENCES

[1] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, Jul. 1997.
[2] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *J. Mach. Learn. Res.*, vol. 6, pp. 615–637, Apr. 2005.
[3] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th ICML*, Helsinki, Finland, 2008, pp. 160–167.
[4] C. Widmer, Y. Altun, N. C. Toussaint, and G. Rätsch, "Inferring latent task structure for multi-task learning via multiple kernel learning," *BMC Bioinformatics*, vol. 11, Suppl. 8, p. S5, Oct. 2010.
[5] G. Li, S. C. H. Hoi, K. Chang, and R. Jain, "Micro-blogging sentiment detection by collaborative online learning," in *IEEE 10th ICDM*, Sydney, NSW, Australia, 2010, pp. 893–898.
[6] G. Li, K. Chang, S. C. H. Hoi, W. Liu, and R. Jain, "Collaborative online learning of user generated content," in *Proc. 20th ACM Int. CIKM*, 2011, pp. 285–290.
[7] N. Cesa-Bianchi, A. Conconi, and C. Gentile, "On the generalization ability of on-line learning algorithms," *IEEE Trans. Inform. Theory*, vol. 50, no. 9, pp. 2050–2057, Sept. 2004.
[8] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *J. Mach. Learn. Res.*, vol. 7, pp. 551–585, Mar. 2006.
[9] K. Crammer and Y. Singer, "Ultraconservative online algorithms for multiclass problems," *J. Mach. Learn. Res.*, vol. 3, pp. 951–991, Jan. 2003.
[10] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958.
[11] L. Yang, R. Jin, and J. Ye, "Online learning by ellipsoid method," in *Proc. 26th Annu. ICML*, Montreal, QC, Canada, 2009, p. 145.
[12] P. Zhao, S. C. H. Hoi, and R. Jin, "DUOL: A double updating approach for online learning," in *NIPS*, 2009, pp. 2259–2267.
[13] C. Gentile, "A new approximate maximal margin classification algorithm," *J. Mach. Learn. Res.*, vol. 2, pp. 213–242, Dec. 2001.
[14] Y. Li and P. M. Long, "The relaxed online maximum margin algorithm," in *NIPS*, 1999, pp. 498–504.
[15] N. Cesa-Bianchi, A. Conconi, and C. Gentile, "A second-order perceptron algorithm," *SIAM J. Comput.*, vol. 34, no. 3, pp. 640–668, Jul. 2005.
[16] M. Dredze, K. Crammer, and F. Pereira, "Confidence-weighted linear classification," in *Proc. 25th ICML*, Helsinki, Finland, 2008, pp. 264–271.
[17] K. Crammer, M. Dredze, and F. Pereira, "Exact convex confidence-weighted learning," in *NIPS*, Dec. 2008, pp. 345–352.
[18] K. Crammer, M. Dredze, and A. Kulesza, "Multi-class confidence weighted algorithms," in *Proc. 2009 Conf. EMNLP*, pp. 496–504.
[19] K. Crammer, A. Kulesza, and M. Dredze, "Adaptive regularization of weight vectors," in *NIPS*, 2009, pp. 414–422.
[20] T. T. Nguyen, K. Chang, and S. C. Hui, "Distribution-aware online classifiers," in *IJCAI*, 2011, pp. 1427–1432.
[21] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *NIPS*, 2006, pp. 41–48.
[22] B. Bakker, "Task clustering and gating for Bayesian multitask learning," *J. Mach. Learn. Res.*, vol. 4, pp. 83–99, Jan. 2003.
[23] S. Ben-David and R. Schuller, "Exploiting task relatedness for mulitple task learning," in *16th Annu. COLT*, Washington, DC, USA, 2003, pp. 567–580.
[24] E. V. Bonilla, K. M. Chai, and C. K. I. Williams, "Multi-task Gaussian process prediction," in *NIPS*, Dec. 2007.
[25] D. Sheldon, "Graphical multi-task learning," in *NIPS'08 Workshop Structured Input Structured Output*, 2008.
[26] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *J. Mach. Learn. Res.*, vol. 6, pp. 1817–1853, Nov. 2005.
[27] O. Chapelle, P. K. Shivaswamy, S. Vadrevu, K. Q. Weinberger, Y. Zhang, and B. L. Tseng, "Multi-task learning for boosting with application to web search ranking," in *Proc. 16th ACM SIGKDD Int. Conf. KDD*, Washington, DC, USA, 2010, pp. 1189–1198.
[28] Z. Kang, K. Grauman, and F. Sha, "Learning with whom to share in multi-task feature learning," in *Proc. 28th ICML*, Bellevue, WA, USA, 2011.
[29] A. Argyriou, C. A. Micchelli, M. Pontil, and Y. Ying, "A spectral regularization framework for multitask structure learning," in *NIPS*, 2008, pp. 25–32.
[30] L. Jacob, F. Bach, and J. Vert, "Clustered multi-task learning," in *NIPS*, 2009, pp. 745–752.
[31] K. Q. Weinberger, A. Dasgupta, J. Langford, A. J. Smola, and J. Attenberg, "Feature hashing for large scale multitask learning," in *Proc. 26th Annu. ICML*, 2009, p. 140.
[32] O. Dekel, P. M. Long, and Y. Singer, "Online multitask learning," in *19th Annu. COLT*, Ft. Lauderdale, FL, USA, 2006, pp. 453–467.
[33] J. Abernethy, P. L. Bartlett, and A. Rakhlin, "Multitask learning with expert advice," in *Proc. 20th Annu. COLT*, 2007, pp. 484–498.
[34] A. Agarwal, A. Rakhlin, and P. Bartlett. (2008, Oct.) "Matrix regularization techniques for online multitask learning," EECS Department, University of California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2008-138 [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-138.html
[35] A. Saha, P. Rai, H. Daumé, III, and S. Venkatasubramanian, "Online learning of multiple tasks and their relationships," in *Proc. 14th Int. Conf. AISTATS*, Ft. Lauderdale, FL, USA, 2011, pp. 643–651.
[36] G. Cavallanti, N. Cesa-Bianchi, and C. Gentile, "Linear algorithms for online multitask classification," *J. Mach. Learn. Res.*, vol. 11, pp. 2901–2934, Oct. 2010.
[37] J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert, "A new approach to collaborative filtering: Operator estimation with spectral regularization," *J. Mach. Learn. Res.* vol. 10, pp. 803–826, Mar. 2009.
[38] J. Zhou, J. Chen, and J. Ye, (2011). *MALSAR: Multi-task Learning via StructurAl Regularization*, Arizona State University, [Online]. Available: http://www.public.asu.edu/~jye02/Software/MALSAR
[39] B. Peters *et al.*, "A community resource benchmarking predictions of peptide binding to MHC-I molecules," *PLoS Comput. Biol.*, vol. 2, no. 6, article. e65, Jun. 2006.
[40] L. Jacob and J.-P. Vert, "Efficient peptide-MHC-i binding prediction for alleles with few known binders," *Bioinformatics*, vol. 24, no. 3, pp. 358–366, Feb. 2008.

Guangxia Li received the bachelor's degree in computer science from Northwestern Polytechnical University, Xi'an, P.R. China, in 2006. He is currently a Ph.D. candidate with the School of Computer Engineering with Nanyang Technological University, Singapore. His current research interests include information retrieval, data mining, statistical machine learning, and statistical and graph models and their applications to real-world problems in text and web mining, sentiment analysis, and social network analysis.

Steven C.H. Hoi is currently an Associate Professor of the School of Computer Engineering at Nanyang Technological University, Singapore. He received the bachelor's degree from Tsinghua University, P.R. China, in 2002, and the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, in 2006. His current research interests include machine learning and its applications to multimedia information retrieval (image and video retrieval), web & social media search and mining, pattern recognition, and computational finance. He has published over 80 referred papers in top conferences and journals in his areas. He has served as General Co-Chair for ACM SIGMM Workshops on Social Media (WSM'09, WSM'10, WSM'11), Program Co-Chair for the 4th Asian Conference on Machine Learning (ACML'12), Book Editor for "Social Media Modeling and Computing", Guest Editor for *ACM Transactions on Intelligent Systems and Technology* (*ACM TIST*), technical program committee member for many international conferences, and external reviewer for many top journals and worldwide funding agencies, including NSF in the U.S., and RGC in Hong Kong.

Kuiyu Chang is an Assistant Professor of computer engineering at Nanyang Technological University, Singapore. Prior to that, he served as a Senior Risk Management Analyst for ClearCommerce (now eFunds). From 2000 to 2002, Kuiyu was a member of the technical staff at Interwoven (now Autonomy). He has served as Program Co-Chair for PAISI (2006-2008), and Publications Chair for PAKDD 2006. Kuiyu has published over 50 papers, included in top journals (*IEEE PAMI*) and conferences (SIGIR, IJCAI, ICDE, ICDM, SDM), and is also the recipient of 2 International Best Paper Awards. Since 2005, he has been the lead on the Review and Opinion Search Engine (ROSE) project, which aggregates online Chinese sentiments. Kuiyu consults regularly for IT companies in China, Singapore, and Malaysia. He received the B.S. degree from National Taiwan University, the M.S. degree from the University of Hawaii at Manoa, and the Ph.D. degree from the University of Texas at Austin, all in electrical/computer engineering.

Wenting Liu received the bachelor's degree in computational mathematics from Wuhan University, P.R. China, in 2006, and the master's degree in software engineering from Beihang University, Beijing, P.R. China, in 2011. She is currently a Ph.D. candidate with the School of Computer Engineering at Nanyang Technological University, Singapore. Her current research interests include data mining, statistical machine learning with applications in bioinformatics.

Ramesh Jain is the Bren Professor of information and computer science, Department of Computer Science, University of California, Irvine, CA, USA. He has been an Active Researcher in multimedia information systems, image databases, machine vision, and intelligent systems. While he was at the University of Michigan, Ann Arbor, MI, USA and the University of California, San Diego, CA, USA, he founded and directed artificial intelligence and visual computing labs. He has co-authored more than 250 research papers. His current research interests include experiential systems and their applications. He was the founding Editor-in-Chief of *IEEE Multimedia Magazine* and *Machine Vision and Applications* and serves on the editorial boards of several magazines in multimedia, business, and image and vision processing. He is a Fellow of ACM, IAPR, AAAI, and SPIE.