OTL: A Framework of Online Transfer Learning

Peilin Zhao ZHAO0106@NTU.EDU.SG
Steven C.H. Hoi CHHOI@NTU.EDU.SG
School of Computer Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798

Abstract

In this paper, we investigate a new machine learning framework called *Online Transfer Learning* (OTL) that aims to transfer knowledge from some source domain to an online learning task on a target domain. We do not assume the target data follows the same class or generative distribution as the source data, and our key motivation is to improve a supervised online learning task in a target domain by exploiting the knowledge that had been learned from large amount of training data in source domains. OTL is in general challenging since data in both domains not only can be different in their class distributions but can be also different in their feature representations. As a first attempt to this problem, we propose techniques to address two kinds of OTL tasks: one is to perform OTL in a homogeneous domain, and the other is to perform OTL across heterogeneous domains. We show the mistake bounds of the proposed OTL algorithms, and empirically examine their performance on several challenging OTL tasks. Encouraging results validate the efficacy of our techniques.

1. Introduction

Transfer learning (TL) has been actively studied recently (Pan & Yang, 2009). It mainly aims to address the machine learning tasks of building models in a new target domain by taking advantage of information from another existing source domain through knowledge transfer. Transfer learning is important for many applications where training data in a new domain may be limited or too expensive to collect. Although transfer learning has been actively explored, most existing work on transfer learning were often studied in an offline learning fashion, which has to assume training data in the new domain is given a priori. Such an assumption may not always hold for some real applications where training examples may arrive in an online/sequential manner.

Unlike the existing transfer learning studies, in this paper, we propose a new framework of **Online Transfer Learning** (OTL), which addresses the transfer learning problem using an online learning framework. As the first attempt to this problem, we address some OTL challenges in two different settings. In the first setting, we study the *homogeneous OTL* where the target domain shares the same feature space as the old/source one. In the second setting, we address the challenge of *heterogeneous OTL* where the feature space of the target domain is different from that of the source domain. We propose algorithms to solve both problems, and theoretically analyze their mistake bounds. Finally, we empirically examine their performance on several challenging OTL tasks.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed framework. Section 4 and Section 5 address the homogeneous and heterogeneous OTL tasks, respectively. Section 6 gives our experimental results and Section 7 concludes this work.

2. Related Work

Our work is generally related to two machine learning topics: *online learning* and *transfer learning*. Below we review some important related work in both areas.

Online learning has been extensively studied for years (Rosenblatt, 1958; Crammer et al., 2006; Zhao et al., 2009; Yang et al., 2010). Unlike typical machine learning methods that assume training examples are available before the learning task, online learning is more appropriate for some real-world problems where training data arrive sequentially. Due to their merits of attractive efficiency and scalability, various online learning methods have been proposed. One well-known approach is the Perceptron algorithm (Rosenblatt, 1958; Freund & Schapire, 1999), which updates the model by adding a new example with some constant weight into the current set of support vectors when the example is misclassified.

Appearing in *Proceedings of the 27^{th} International Conference on Machine Learning*, Haifa, Israel, 2010. Copyright 2010 by the author(s)/owner(s).

Recently many online learning algorithms have been proposed based on the criterion of maximum margin (Crammer et al., 2006; Li & Long, 1999; Zhao et al., 2009). One example is the Passive-Aggressive (PA) method (Crammer et al., 2006), which updates the classification model when a new example is misclassified or its classification score is smaller than some predefined margin. More extensive surveys for online learning can be found in (Shalev-Shwartz, 2007).

Transfer learning (TL) has been actively studied. The goal of TL is to extract knowledge from one or more source tasks and then apply them to a target task. Various TL methods have been proposed. According to different learning types, these methods can be roughly classified into three categories: *inductive*, *transductive*, and *unsupervised* approaches. Inductive TL (DaumáIII & Marcu, 2006) aims to induce the model in the target domain with the aid of knowledge transferred from the source domains; transductive TL (Arnold et al., 2007) aims to extract the knowledge from source domain to improve the prediction tasks in the target domain without labeled data in the target domain; while unsupervised TL aims to resolve unsupervised learning tasks in target domain (Dai et al., 2008). Moreover, according to different feature representation, TL can be classified as *homogeneous* vs. *heterogeneous* TL (Argyriou et al., 2008) where the feature spaces of source and target domains can be different. A comprehensive survey on transfer learning can be found in (Pan & Yang, 2009).

Although both online learning and transfer learning have been actively studied, to the best of our knowledge, no existing work has formally addressed transfer learning by an online learning framework. Finally, we note that OTL is also different from *online multi-task learning* (Dekel et al., 2007), which aims to learn multiple tasks in parallel in an online learning framework.

3. Problem Formulation

Let us denote by $\mathcal{X}_1 \times \mathcal{Y}_1$ the source/old data space, where $\mathcal{X}_1 = \mathbb{R}^m$ and $\mathcal{Y}_1 = \{-1, +1\}$. Since our task aims to learn a kernel classifier, we thus denote by $\kappa_1(\cdot,\cdot) : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$ the kernel function to be used in the source classifier. Assume that a source classifier $h(x)$ can be represented as:

$$h(x) = \sum_{s=1}^{S} \alpha_s y_{1_s} \kappa_1(x_{1_s}, x)$$

where $\{(x_{1_s}, y_{1_s}) \in \mathcal{X}_1 \times \mathcal{Y}_1 | s = 1, \ldots S\}$ are the set of support vectors for the source training data set, and α_s are the coefficients of support vectors. Typically the source classifier $h(x)$ can be obtained by applying existing learning techniques, such as online learning via the Perceptron algorithm (Rosenblatt, 1958; Freund & Schapire, 1999) or regular learning by support vector machines (SVM).

For an online transfer learning (OTL) task, our goal is to online learn some prediction function $f \in \mathcal{H}_\kappa$ on a target domain from a sequence of examples $\{(x_{2_t}, y_{2_t}) | t = 1, \ldots, T\}$ in some data space $\mathcal{X}_2 \times \mathcal{Y}_2$.

Specifically, during the OTL task, at the t-th trial of online learning task, the learner receives an instance x_{2_t}, and the goal of online learning is to find a good prediction function such that the predicted class label $sign(f_t(x_{2_t}))$ can match its truth class label y_{2_t}. The key challenge of OTL is how to effectively transfer the knowledge from the old/source domain to the new/target domain for improving the online learning performance. Next, we study OTL in two different settings: homogeneous vs. heterogeneous OTL.

4. Online Transfer Learning over Homogeneous Domains

We start by studying the homogeneous OTL, in which we assume the source domain and the target domain have the same feature space, i.e., $\mathcal{X}_2 = \mathcal{X}_1$ and $\mathcal{Y}_2 = \mathcal{Y}_1$. One key challenge of this task is to address the *concept drifting* issue that often occurs in this scenario. Specifically, the concept drift means that the target variable to be predicted changes over time in the learning process. This raises the challenge of transferring knowledge from source domain to target domain.

The basic idea of our OTL solution is based on the ensemble learning approach. In particular, we first construct an entirely new prediction function f only from the data in the target domain in an *online* fashion, and then learn an ensemble prediction function that is the mixture of both the old and the new prediction functions, i.e., h and f, which thus can transfer the knowledge from the source domain. The remaining issue is then how to effectively combine the two prediction functions for handling the concept drift issue.

To combine the two prediction functions $h(x)$ and $f_t(x)$ at the t-trial of the online learning task, we introduce two weight parameters, $w_{1,t}$ and $w_{2,t}$, for the two prediction functions respectively. At the t-th step, given an instance x_{2_t}, we predict its class label by the following ensemble function:

$$\hat{y}_{2_t} = \text{sign}\Big(w_{1,t}\Pi(h(x_{2_t})) + w_{2,t}\Pi(f_t(x_{2_t})) - \frac{1}{2}\Big) \qquad (1)$$

where $\Pi(x)$ is a normalization function, i.e., $\Pi(x) = \max(0, \min(1, \frac{x+1}{2}))$. At the beginning of the OTL task, we simply set $w_{1,1} = w_{2,1} = \frac{1}{2}$. In order to effectively transfer, for the subsequent trials of the

```
Algorithm 1: Online Transfer Learning algorithm (OTL)
INPUT: the old classifier $h(x) = \sum_{s=1}^{S} \alpha_s y_{1_s} \kappa_1(x_{1_s}, x)$ and
initial trade off $C$ and weights $w_{1,1} = w_{2,1} = \frac{1}{2}$
Initialize $f_1 = 0$
for $t = 1, 2, \ldots, T$ do
    receive instance: $x_{2_t} \in X_2$
    predict $\hat{y}_{2_t}$ by Eq. 1
    receive correct label: $y_{2_t} \in \{-1, +1\}$
    compute $w_{1,t+1}$ and $w_{2,t+1}$ by Eq. (2) and (3)
    suffer loss: $\ell_t = [1 - y_{2_t} f_t(x_{2_t})]_+$
    if $\ell_t > 0$ then
        $\tau_t = \min\{C, \ell_t / \kappa_2(x_{2_t}, x_{2_t})\}$
        $f_{t+1} = f_t + \tau_t y_{2_t} \kappa_2(x_{2_t}, \cdot)$
    end if
end for
```

Figure 1. The Online Transfer Learning (OTL) algorithm.

OTL task, in addition to updating the function $f_{t+1}(x)$ by some online learning methods, e.g. the PA algorithm (Crammer et al., 2006), we expect the two weights of both prediction functions, i.e., $w_{1,t}$ and $w_{2,t}$, can be adjusted dynamically. We suggest the following updating scheme for adjusting the weights:

$$w_{1,t+1} = \frac{w_{1,t} * s_t(h)}{w_{1,t} * s_t(h) + w_{2,t} * s_t(f_t)} \qquad (2)$$

$$w_{2,t+1} = \frac{w_{2,t} * s_t(f_t)}{w_{1,t} * s_t(h) + w_{2,t} * s_t(f_t)} \qquad (3)$$

where $s_t(g) = \exp\{-\eta \ell^*(\Pi(g(x_{2_t})), \Pi(y_{2_t}))\}, \quad \forall g \in \mathcal{H}_\kappa$ and $\ell^*(z, y)$ is a loss function which is set to $\ell^*(z, y) = (z - y)^2$ in our approach. Finally, Figure 1 summarizes the proposed OTL algorithm.

Next we analyze the mistake bound of the algorithm. We first introduce the following proposition.

Proposition 1. *When using the square loss $\ell^*(z, y) = (z - y)^2$ for $z \in [0, 1]$ and $y \in \{0, 1\}$ and the above exponentially weighting update method and setting $\eta = 1/2$, we have the bound of the ensemble algorithm as:*

$$\sum_{t=1}^{T} \ell^*(w_{1,t}\Pi(h(x_{2_t})) + w_{2,t}\Pi(f_t(x_{2_t})), \Pi(y_{2_t})) \leq 2\ln 2 +$$

$$\min\left\{\sum_{t=1}^{T} \ell^*(\Pi(h(x_{2_t})), \Pi(y_{2_t})), \sum_{t=1}^{T} \ell^*(\Pi(f_t(x_{2_t})), \Pi(y_{2_t}))\right\}$$

The proposition can be proved by following the similar technique described at Section 3.3 of the book (Cesa-Bianchi & Lugosi, 2006). By Proposition 1, we derive the mistake bound of the OTL algorithm as follows.

Theorem 1. *Let us denote by M the number of mistakes made by the OTL algorithm, we then have M bounded from above by:*

$$M \leq 4 \min\left\{\Sigma_h, \Sigma_f\right\} + 8\ln 2 \qquad (4)$$

where $\Sigma_h = \sum_{t=1}^{T} \ell^*(\Pi(h(x_{2_t})), \Pi(y_{2_t}))$ and $\Sigma_f = \sum_{t=1}^{T} \ell^*(\Pi(f_t(x_{2_t})), \Pi(y_{2_t}))$.

The proof of Theorem 1 is given in the appendix.

Remark. To better understand the mistake bound, we denote by M_h and M_f the mistake bound of model h and f_t, respectively. First, we note that $\ell^*(\Pi(h(x_{2_t})), \Pi(y_{2_t}))$ is the upper bound of $\frac{1}{4}M_h$ instead of M_h (because ℓ is a square loss and both $\Pi(h(x_{2_t}))$ and $\Pi(y_{2_t})$ are normalized to $[0, 1]$); similarly, $\ell^*(\Pi(f_t(x_{2_t})), \Pi(y_{2_t}))$ is the upper bound of $\frac{1}{4}M_f$. Further, if we assume $\ell^*(\Pi(h(x_{2_t})), \Pi(y_{2_t})) \approx \frac{1}{4}M_h$ and $\ell^*(\Pi(f_t(x_{2_t})), \Pi(y_{2_t})) \approx \frac{1}{4}M_f$, we have the result: $M \leq \min\{M_h, M_f\} + 8\ln 2$. This gives a strong theoretical support for the OTL algorithm.

5. Online Transfer Learning over Heterogeneous Domains

In this section, we study the OTL problem across heterogeneous domains where the source and target domains have different feature spaces.

Heterogeneous OTL is generally very challenging. To simplify the problem, we assume the feature space of the source domain is a subset of that of the target domain. Due to the difference of the two feature spaces, we cannot directly apply the algorithm in the previous section. Below we propose to introduce a multi-view approach for solving the challenge in this case.

Formally, we denote the data on the target domain as: $\{(x_{2_t}, y_{2_t}) | t = 1, \ldots, T\}$, where $x_{2_t} \in \mathcal{X}_2 = \mathbb{R}^n \supset \mathbb{R}^m$ and $y_{2_t} \in \{-1, +1\}$. Without loss of generality, we assume the first m dimensions of $\mathcal{X}_2$ represent the old feature space $\mathcal{X}_1$. In the multi-view setting, we split each data instance x_{2_t} into two instances $x_{2_t}^{(1)} \in X_1$ and $x_{2_t}^{(2)} \in \mathcal{X}_2 / \mathcal{X}_1$. For the second view, we introduce a new kernel function $\kappa_2(\cdot, \cdot) : \mathbb{R}^{n-m} \times \mathbb{R}^{n-m} \to \mathbb{R}$.

The key idea of our heterogeneous OTL method is to adopt a co-regularization principle of online learning two classifiers $f_t^{(1)}$ and $f_t^{(2)}$ simultaneously from the two views, and predict an unseen example on the target domain by $\hat{y}_t = sign\left(\frac{1}{2}\left(f_t^{(1)}(x_{2_t}^{(1)}) + f_t^{(2)}(x_{2_t}^{(2)})\right)\right)$.

For the specific algorithm, we initialize the classifier for the first view by setting $f_1^{(1)} = h$, and setting $f_1^{(2)} = 0$ for the second view. For a new example in the online learning task, we update the new functions $f_{t+1}^{(1)}$ and $f_{t+1}^{(2)}$ by the following *co-regularization* optimization:

$$(f_{t+1}^{(1)}, f_{t+1}^{(2)}) = \arg \min_{f^{(1)} \in \mathcal{H}_{\kappa_1} f^{(2)} \in \mathcal{H}_{\kappa_2}} \frac{\gamma_1}{2}\|f^{(1)} - f_t^{(1)}\|_{\mathcal{H}_{\kappa_1}}^2 + \frac{\gamma_2}{2}\|f^{(2)} - f_t^{(2)}\|_{\mathcal{H}_{\kappa_2}}^2 + C\ell_t \qquad (5)$$

where γ_1, γ_2 and C are positive parameters, and the loss term ℓ_t is defined below:

$$\ell_t = [1 - y_{2_t}\frac{1}{2}(f^{(1)}(x_{2_t}^{(1)}) + f^{(2)}(x_{2_t}^{(2)}))]_+ \qquad (6)$$

```
Algorithm 2: The Co-Regularized Online Transfer Learning Algorithm (COTL)
INPUT: the old classifier $h(x) = \sum_{s=1}^{S} \alpha_s y_{1_s} \kappa_1(x_{1_s}, x)$ and parameters $\gamma_1$, $\gamma_2$ and $C$
Initialize $f_1^{(1)} = h$ and $f_1^{(2)} = 0$
for $t = 1, 2, \ldots, T$ do
    receive instance: $x_{2_t} \in \mathcal{X}_2$
    predict: $\hat{y}_t = sign\left(\frac{1}{2}\left(f_t^{(1)}(x_{2_t}^{(1)}) + f_t^{(2)}(x_{2_t}^{(2)})\right)\right)$
    receive correct label: $y_{2_t} \in \{-1, +1\}$
    suffer loss:
    $\ell_t = \left[1 - y_{2_t}\left(\frac{1}{2}\left(f_t^{(1)}(x_{2_t}^{(1)}) + f_t^{(2)}(x_{2_t}^{(2)})\right)\right)\right]_+$
    if $\ell_t > 0$ then
        $\tau_t = \min\{C, \frac{4\gamma_1\gamma_2\ell_t}{k_t^1\gamma_2 + k_t^2\gamma_1}\}$
        $f_{t+1}^{(1)} = f_t^{(1)} + \frac{\tau_t}{2\gamma_1} y_{2_t} \kappa_1(x_{2_t}^{(1)}, \cdot)$
        $f_{t+1}^{(2)} = f_t^{(2)} + \frac{\tau_t}{2\gamma_2} y_{2_t} \kappa_2(x_{2_t}^{(2)}, \cdot)$
    end if
end for
```

Figure 2. The Co-regularized Online Transfer Learning.

Intuitively, the above updating method aims to make the updated ensemble classifier be able to classify the new observed example (x_{2_t}, y_{2_t}) correctly, and to force the two-view classifiers without deviating too much from the previous classifiers $(f_t^{(1)}, f_t^{(2)})$ via the first two regularization terms.

The above optimization enjoys a closed-form solution as shown in Proposition 2. To simplify our discussion, we introduce notation $k_t^1 = \kappa_1(x_{2_t}^{(1)}, x_{2_t}^{(1)})$ and $k_t^2 = \kappa_2(x_{2_t}^{(2)}, x_{2_t}^{(2)})$.

Proposition 2. *For the optimization problem (5), its solution can be expressed as follows:*

$$f_{t+1}^{(i)} = f_t^{(i)} + \frac{\tau_t}{2\gamma_i}\kappa_i(x_{2_t}^{(i)}, \cdot) \quad i = 1, 2 \tag{7}$$

where $\tau_t = \min\{C, \frac{4\gamma_1\gamma_2\ell_t}{k_t^1\gamma_2 + k_t^2\gamma_1}\}$.

The proof of the proposition is given in the appendix.

By this proposition, we summarize the proposed "Co-regularized Online Transfer Learning" (COTL) algorithm in Figure 2.

Before we prove the mistake bound for the COTL algorithm, we first introduce a lemma.

Lemma 1. *Let $(x_{2_t}, y_{2_t}), t = 1, \ldots, T$ be a sequence of examples, where $x_{2_t} \in \mathbb{R}^n$ and $y_{2_t} \in \{-1, +1\}$ for all t. After we split the instance x_{2_t} into two views $(x_{2_t}^{(1)}, x_{2_t}^{(2)})$, for any $g^{(1)} \in \mathcal{H}_{\kappa_1}$ and $g^{(2)} \in \mathcal{H}_{\kappa_2}$, we have the following bound:*

$$\begin{aligned}&\sum_{t=1}^{T}\tau_t\left(\ell_t - \ell(g^{(1)}, g^{(2)}; t) - (\frac{k_t^1}{8\gamma_1} + \frac{k_t^2}{8\gamma_2})\tau_t\right)\\ &\leq \frac{\gamma_1}{2}\|h - g^{(1)}\|^2 + \frac{\gamma_2}{2}\|g^{(2)}\|^2\end{aligned} \tag{8}$$

where ℓ_t is given in Eqn. (6) and ℓ is defined as: $\ell(g^{(1)}, g^{(2)}; t) = [1 - y_{2_t}\frac{1}{2}(g^{(1)}(x_{2_t}^{(1)}) + g^{(2)}(x_{2_t}^{(2)}))]_+$.

The proof of the Lemma is given in the appendix. Using Lemma 1, we can show the following theorem for the mistake bound of the proposed COTL algorithm.

Theorem 2. *Let $(x_{2_t}, y_{2_t}), t = 1, \ldots, T$ be a sequence of examples, where $x_{2_t} \in \mathbb{R}^n$ and $y_{2_t} \in \{-1, +1\}$ for all t. In addition $k_t^1 \leq R_1$ and $k_t^2 \leq R_2$ $t = 1, \ldots, T$. And we split the instance x_{2_t} into two views $(x_{2_t}^{(1)}, x_{2_t}^{(2)})$. Then for any $g^{(1)} \in \mathcal{H}_{\kappa_1}$ and $g^{(2)} \in \mathcal{H}_{\kappa_2}$, the number of mistakes M made by the proposed COTL algorithm is bounded from above by:*

$$M \leq \frac{1}{\tau}\Big(\gamma_1\|h - g^{(1)}\|^2 + \gamma_2\|g^{(2)}\|^2 + 2C\sum_{t=1}^{T}\ell(g^{(1)}, g^{(2)}; t)\Big)$$

where $\tau = \min\{C, \frac{4\gamma_1\gamma_2}{R_1\gamma_2 + R_2\gamma_1}\}$.

Proof. Since $\tau_t = \min\{C, \frac{4\gamma_1\gamma_2\ell_t}{k_t^1\gamma_2 + k_t^2\gamma_1}\} \leq C$, $\tau_t\ell(g^{(1)}, g^{(2)}; t) \leq C\ell(g^{(1)}, g^{(2)}; t)$. In addition, $\tau_t = \min\{C, \frac{4\gamma_1\gamma_2\ell_t}{k_t^1\gamma_2 + k_t^2\gamma_1}\} \leq \frac{4\gamma_1\gamma_2\ell_t}{k_t^1\gamma_2 + k_t^2\gamma_1}$, we thus have

$$\begin{aligned}&\sum_{t=1}^{T}\tau_t\left(\ell_t - \ell(g^{(1)}, g^{(2)}; t) - (\frac{k_t^1}{8\gamma_1} + \frac{k_t^2}{8\gamma_2})\tau_t\right)\\ &= \sum_{t=1}^{T}\tau_t\ell_t - \sum_{t=1}^{T}\tau_t\ell(g^{(1)}, g^{(2)}; t) - \sum_{t=1}^{T}(\frac{k_t^1}{8\gamma_1} + \frac{k_t^2}{8\gamma_2})\tau_t^2\\ &\geq \sum_{t=1}^{T}\tau_t\ell_t - \sum_{t=1}^{T}C\ell(g^{(1)}, g^{(2)}; t) - \sum_{t=1}^{T}(\frac{k_t^1}{8\gamma_1} + \frac{k_t^2}{8\gamma_2})\tau_t\frac{4\gamma_1\gamma_2\ell_t}{k_t^1\gamma_2 + k_t^2\gamma_1}\\ &= \sum_{t=1}^{T}\tau_t\ell_t - C\sum_{t=1}^{T}\ell(g^{(1)}, g^{(2)}; t) - \frac{1}{2}\sum_{t=1}^{T}\tau_t\ell_t\\ &= \frac{1}{2}\sum_{t=1}^{T}\tau_t\ell_t - C\sum_{t=1}^{T}\ell(g^{(1)}, g^{(2)}; t)\end{aligned}$$

Combining the above inequality with the conclusion of Lemma 1, we have

$$\frac{1}{2}\sum_{t=1}^{T}\tau_t\ell_t \leq \frac{\gamma_1}{2}\|h - g^{(1)}\|^2 + \frac{\gamma_2}{2}\|g^{(2)}\|^2 + C\sum_{t=1}^{T}\ell(g^{(1)}, g^{(2)}; t)$$

Furthermore, when a mistake occurs, $\ell_t \geq 1$; thus $\tau_t\ell_t = \min\{C, \frac{4\gamma_1\gamma_2\ell_t}{k_t^1\gamma_2 + k_t^2\gamma_1}\} * \ell_t \geq \min\{C, \frac{4\gamma_1\gamma_2\ell_t}{k_t^1\gamma_2 + k_t^2\gamma_1}\} \geq \min\{C, \frac{4\gamma_1\gamma_2}{R_1\gamma_2 + R_2\gamma_1}\} = \tau$. Combining this observation with the inequality above, we have

$$\frac{1}{2}M \times \tau \leq \frac{\gamma_1}{2}\|h - g^{(1)}\|^2 + \frac{\gamma_2}{2}\|g^{(2)}\|^2 + C\sum_{t=1}^{T}\ell(g^{(1)}, g^{(2)}; t)$$

The theorem follows directly by multiplying $2/\tau$ on both sides of the above inequality. □

6. Experimental Results

In this section, we evaluate the empirical performance of the proposed two kinds of OTL algorithms.

6.1. Experimental Testbed and Setup for Homogeneous OTL

Our first experiment is to evaluate the performance of OTL from homogeneous data. We compare our OTL technique against other popular online learning techniques, including the Passive-Aggressive algorithms("PA") (Crammer et al., 2006) without exploiting any knowledge from the source domain, and a variant of it, which is the **PA** method **I**nitialized with the **O**ld classifier h, denoted as **PAIO** for short. For our OTL technique, in addition to Algorithm 1, we also implement another variant, which is implemented by fixing the ensemble weights of the OTL algorithm to 1/2, denoted "OTL(fixed)" for short. This helps us to examine if the proposed weighting strategy is effective. For the PA methods, the original algorithm was proposed for learning linear models (Crammer et al., 2006). In our experiments, we adapted all the algorithms to the kernel settings.

To extensively examine the performance, we test all the algorithms on some benchmark machine learning datasets, including dataset "w7a", a dataset without concept drifting, and "usenet2", a dataset with concept-drifting, which can be downloaded [1]. Besides, we also create another concept-drifting dataset named "newsgroup4" based on the dataset "newsgroup20" downloaded from the LIBSVM web site [2]. The details of the "newsgroup4" is shown in Table 1. For an OTL experiment, we must split the whole dataset into two parts: (1) training data for the source domain, and (2) test data for online learning in the target domain. For the two concept-drifting datasets, we split each of them into two parts according to their sequential orders: "usenet2"(300+1200) and "newsgroup4"(400+1200); for "w7a" without concept drifting, we randomly split it into two parts: (10000+14692), and repeat it 20 times. Finally, we adopt the (kernel) PA algorithm to build the baseline classifier in the source domain.

Table 1. The class distribution of dataset *newsgroup4*.

example id	0-400	401-800	801-1200	1201-1600
comp.windows.x	+	-	-	+
rec.sport.hockey	+	+	-	-
sci.space	-	+	+	-
talk.politics.mideast	-	-	+	+

All the algorithms employ a gaussian kernel. For fair comparison and simplicity, we set $\sigma_1 = 4$ and $\sigma_2 = 8$ for all the datasets and algorithms. In addition, parameter C is set to 5 for all algorithms on every dataset. We evaluate the performance of online learning methods by measuring the standard mistake rate. Also we evaluate the total number of support vectors to examine the sparsity of the resulting classifiers. Finally, we measure the average time cost for comparing the efficiency of the algorithms.

6.2. Evaluation of Homogeneous OTL Tasks

Table 2 summarizes the performance of the compared algorithms on the three datasets.

Table 2. Results on the datasets of homogeneous domain.

Algorithm	w7a (n=24692, d=300)		
	Mistake (%)	Support Vectors (#)	Time (s)
PA	2.86 %± 0.05	1639.95 ± 23.96	0.96
PAIO	2.34 %± 0.06	2556.20 ± 30.95	1.82
OTL(fixed)	2.22 %± 0.05	3045.75 ± 33.99	2.22
OTL	**1.87 %± 0.01**	3045.75 ± 33.99	2.37

Algorithm	usenet2 (n=1500, d=99)		
	Mistake (%)	Support Vectors (#)	Time (s)
PA	49.33 %± 0	949 ± 0	0.04
PAIO	47.92 %± 0	1116 ± 0	0.04
OTL(fixed)	42.67 %± 0	1203 ± 0	0.05
OTL	**34.42 %± 0**	1203 ± 0	0.07

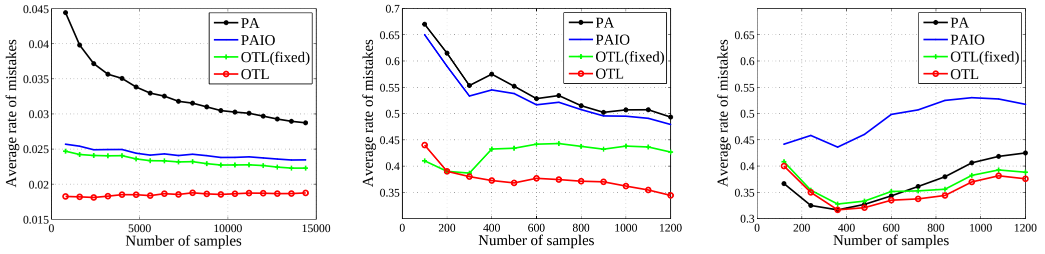
Algorithm	newsgroup4 (n=1600, d=62062)		
	Mistake (%)	Support Vectors (#)	Time (s)
PA	42.50 %± 0	1188 ± 0	0.04
PAIO	51.75 %± 0	1585 ± 0	0.05
OTL(fixed)	38.83 %± 0	1536 ± 0	0.06
OTL	**37.58 %± 0**	1536 ± 0	0.08

Several observations can be drawn from the experimental results. First of all, for the "w7a" dataset without concept drifting, we found that all three algorithms outperform the baseline algorithm (PA), in which the proposed OTL algorithm achieved the best performance among all. Further, on the two concept-drifting datasets, we found that the performances of the compared algorithms are quite different. For PAIO, it can only improve the mistake rate slightly on the "usenet2", but failed to improve over the baseline on the dataset "newsgroup4". For the two OTL algorithms, both can improve the mistake rates on both datasets, in which OTL is more effective than OTL(fixed) which uses a fixed combination weights. These experimental results show that without careful consideration, OTL may suffer from negative transfer when facing a serious concept drifting problem.

Finally, Figure 3 shows the details of average mistake rates varying over the OTL processes on the three data sets, respectively. Similar to the previous results, the proposed OTL algorithm achieved the best results among all datasets. In particular, on the *newsgroup4* dataset, we found that all the three algorithms suffer from the concept-drifting event at the very beginning of the OTL process; however, the proposed OTL algorithm is able to rapidly improve its performance when receiving more examples, while PAIO failed to improve since it depends too much on the knowledge inherited from the source domain. This again verifies the efficacy of the proposed method.

[1] http://mlkd.csd.auth.gr/concept_drift.html

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools



(a) *w7a* (b) *usenet2* (c) *newsgroup4*

Figure 3. Experimental results of average mistake rates on the homogeneous OTL tasks.

6.3. Experimental Testbed and Setup for Heterogenous OTL

In this section, we evaluate the empirical performance of the proposed Co-regularized Online Transfer Learning (COTL) algorithm for heterogenous OTL tasks. We compare our COTL technique with the PA algorithm, which does not exploit knowledge from the source domain. Similarly, we implement a variant of PA algorithm that uses only the first view of the data and is initialized with h from the source domain, denoted as “PAIO”. We also implement a variant of COTL, whose first view classifier is initialized with zero function, denoted as ”COTL0”. This method enables us to examine the importance of engaging the h function learned from the source domain. Finally, we implement another baseline algorithm that simply combines the above PAIO classifier of the first view and the PA classifier learned from the second view on the target domain, denoted as “SCT” for short.

To extensively examine the performance, we test all the algorithms on several benchmark datasets from machine learning repositories, including “a7a”, “german”, “mushrooms”, “spambase” and “w7a”. These datasets can be downloaded from LIBSVM website.

Table 3. Summary of data sets used for OTL tasks.

Datasets	source/old domain		target/new domain	
	Number	Dimension	Number	Dimension
a7a	6100	61	10000	123
mushrooms	3000	56	5124	112
spambase	2000	28	2601	57
w7a	10000	150	14692	300

For each dataset, we randomly split it into two parts: *source* versus *target*, as shown in Table 3. In the partition, to meet the setup of the heterogenous OTL task, the source-domain data associate with only the first half of the feature space while the target-domain data include the whole feature space.

All the algorithms in comparison employ a gaussian kernel. For fair comparison and simplicity, for all the datasets and algorithms, we set $\gamma_1 = \gamma_2 = 1$ and $\sigma_1 = \sigma_2 = 4$ for the two views, and $\sigma = 8$ for the whole feature. In addition, parameter C is set to 5 for all the algorithms on every dataset. We conducted 20 different random permutations to obtain the average results. We evaluate the performance of online learning methods by calculating the mistake rates. We also evaluate the total number of support vectors to examine the sparsity of the resulting classifiers. Finally, we evaluate the time cost of the compared algorithms.

6.4. Evaluation of Heterogenous OTL Tasks

Table 4 summarizes the performance of all the algorithms for heterogenous OTL on the four datasets.

Table 4. Results on the datasets of heterogenous domain.

Algorithm	a7a		
	Mistake (%)	Support Vectors (#)	Time (s)
PA	22.08 %± 0.34	4266.05 ± 47.10	1.49
PAIO	22.60 %± 0.31	7115.00 ± 33.15	3.55
COTL0	21.62 %± 0.29	8403.20 ± 86.34	3.25
SCT	26.06 %± 0.36	11842.20 ± 56.15	6.16
COTL	**21.32 %± 0.27**	11063.30 ± 87.58	5.71

Algorithm	mushrooms		
	Mistake (%)	Support Vectors (#)	Time (s)
PA	2.56 %± 0.11	944.30 ± 17.93	0.23
PAIO	0.66 %± 0.07	865.75 ± 10.94	0.24
COTL0	1.10 %± 0.09	1450.30 ± 36.68	0.36
SCT	1.10 %± 0.10	2198.35 ± 26.22	0.53
COTL	**0.38 %± 0.04**	1779.90 ± 34.63	0.47

Algorithm	spambase		
	Mistake (%)	Support Vectors (#)	Time (s)
PA	25.04 %± 0.66	1761.30 ± 16.40	0.17
PAIO	14.41 %± 0.57	1742.65 ± 13.46	0.21
COTL0	12.78 %± 0.34	2399.30 ± 24.97	0.25
SCT	12.71 %± 0.40	3667.30 ± 23.89	0.40
COTL	**11.17 %± 0.43**	2993.10 ± 27.49	0.34

Algorithm	w7a		
	Mistake (%)	Support Vectors (#)	Time (s)
PA	3.85 %± 0.08	1780.60 ± 23.28	0.96
PAIO	3.53 %± 0.05	2774.45 ± 32.96	1.91
COTL0	3.34 %± 0.07	3432.00 ± 43.78	1.89
SCT	3.22 %± 0.06	4427.85 ± 39.33	3.03
COTL	**3.04 %± 0.06**	4480.50 ± 57.85	3.08

Several observations can be drawn from the results. First of all, we found that among all the algorithms, the PA algorithm without exploiting knowledge from source domain achieved the highest mistake rate in most cases. This shows that it is important for studying knowledge transfer in an OTL task. Second, for all the datasets, we found that the COTL algorithm has the smallest mistake rate. This validates the proposed OTL technique is effective for knowledge transfer in the online learning tasks. Of course, there is some cost of knowledge transfer for the gain. By examining the number of support vectors and the running time cost, we found that the COTL techniques usually produce denser classifiers and spend more time. This is unavoidable as the COTL algorithm makes use of the old classifier from the source domain. Finally, Figure 4 shows the details of the COTL processes on the four data sets, respectively. Similar observations can be found from the results, which again verify the proposed OTL method is effective and promising.

7. Conclusion

In this paper, we studied the new problem of **Online Transfer Learning** (OTL), which aims to transfer knowledge from a source domain to an online learning task on a target domain. We addressed two OTL tasks in different settings and presented two novel OTL algorithms. We offered theoretical analysis on the mistake bounds of the proposed OTL algorithms, and extensively examined their empirical performance. Encouraging results show the proposed algorithms are effective. Through this work, we hope to encourage the investigation of OTL to address other harder problems, e.g. how to perform heterogeneous OTL from complex data of completely diverse feature representations.

Appendix

Proof of Theorem 1

Proof. First notice that whenever there is a mistake at some t-th step, we should have $|w_{1,t}\Pi(h(x_{2_t})) + w_{2,t}\Pi(f_t(x_{2_t})) - \Pi(y_{2_t})| \geq \frac{1}{2}$. Thus, we haves

$$\sum_{t=1}^{T} \ell^*\left(w_{1,t}\Pi(h(x_{2_t})) + w_{2,t}\Pi(f_t(x_{2_t})), \Pi(y_{2_t})\right)$$
$$= \sum_{t=1}^{T} \left(w_{1,t}\Pi(h(x_{2_t})) + w_{2,t}\Pi(f_t(x_{2_t})) - \Pi(y_{2_t})\right)^2 \geq \frac{1}{4}M$$

Combining the above fact with Proposition 1, we have

$$\frac{1}{4}M \leq \min\left\{\Sigma_h, \Sigma_f\right\} + 2\ln 2$$

where $\Sigma_h = \sum_{t=1}^{T} \ell^*(\Pi(h(x_{2_t})), \Pi(y_{2_t}))$ and $\Sigma_f = \sum_{t=1}^{T} \ell^*(\Pi(f_t(x_{2_t})), \Pi(y_{2_t}))$. The theorem follows directly by multiplying 4 at both sides of the above inequality. □

Proof of Proposition 2

Proof. It is easy to see that the optimization problem (5) is equivalent to the following problem

$$\min_{f^{(1)}\in\mathcal{H}_{\kappa_1} f^{(2)}\in\mathcal{H}_{\kappa_2}} \frac{\gamma_1}{2}\|f^{(1)} - f_t^{(1)}\|^2_{\mathcal{H}_{\kappa_1}} + \frac{\gamma_2}{2}\|f^{(2)} - f_t^{(2)}\|^2_{\mathcal{H}_{\kappa_2}} + C\xi$$
$$s.t. \quad 1 - y_{2_t}\frac{1}{2}\left(f^{(1)}(x_{2_t}^{(1)}) + f^{(2)}(x_{2_t}^{(2)})\right) \leq \xi \quad and \quad \xi \geq 0$$

The Lagrangian of the above optimization is:

$$\begin{aligned}
&\mathcal{L}(f^{(1)}, f^{(2)}, \xi, \tau_t, \lambda)\\
&= \frac{\gamma_1}{2}\|f^{(1)} - f_t^{(1)}\|^2_{\mathcal{H}_{\kappa_1}} + \frac{\gamma_2}{2}\|f^{(2)} - f_t^{(2)}\|^2_{\mathcal{H}_{\kappa_2}} + C\xi\\
&\quad +\tau_t\left(1 - y_{2_t}\frac{1}{2}\left(f^{(1)}(x_{2_t}^{(1)}) + f^{(2)}(x_{2_t}^{(2)})\right) - \xi\right) - \lambda\xi\\
&= \frac{\gamma_1}{2}\|f^{(1)} - f_t^{(1)}\|^2_{H_{\kappa_1}} + \frac{\gamma_2}{2}\|f^{(2)} - f_t^{(2)}\|^2_{H_{\kappa_2}} + \xi(C\\
&\quad -\tau_t - \lambda) + \tau_t\left(1 - y_{2_t}\frac{1}{2}\left(f^{(1)}(x_{2_t}^{(1)}) + f^{(2)}(x_{2_t}^{(2)})\right)\right) \qquad (9)
\end{aligned}$$

where $\tau_t \geq 0$ and $\lambda \geq 0$ are Lagrange multipliers. We now find the minimum of the Lagrangian with respect to $f^{(1)}$, $f^{(2)}$ and ξ by setting their partial derivatives to zeros. We get $f^{(i)} = f_t^{(i)} + \frac{\tau_t}{2\gamma_i}y_{2_t}\kappa_i(x_{2_t}^{(i)}, \cdot)$ for $i = 1, 2$ and $C - \tau_t - \lambda = 0$. And since $\lambda \geq 0$, we conclude $C \geq \tau_t$. We thus have $\tau_t \in [0, C]$.

Plugging the three equations $f^{(i)} = f_t^{(i)} + \frac{\tau_t}{2\gamma_i}y_{2_t}\kappa_i(x_{2_t}^{(i)}, \cdot)$ (where $i = 1, 2$) and $C - \tau_t - \lambda = 0$ into Eq. (9), we have

$$\mathcal{L}(\tau_t) = -\tau_t^2(\frac{k_t^1}{8\gamma_1} + \frac{k_t^2}{8\gamma_2}) + \tau_t\ell_t$$

By setting the partial derivative of the above equation to zero, we have

$$\tau_t = \ell_t/(\frac{k_t^1}{4\gamma_1} + \frac{k_t^2}{4\gamma_2}) = \frac{4\gamma_1\gamma_2\ell_t}{k_t^1\gamma_2 + k_t^2\gamma_1}$$

Finally, combining the result $\tau_t \in [0, C]$, we thus have the solution: $\tau_t = \min\{C, \frac{4\gamma_1\gamma_2\ell_t}{k_t^1\gamma_2+k_t^2\gamma_1}\}$. □

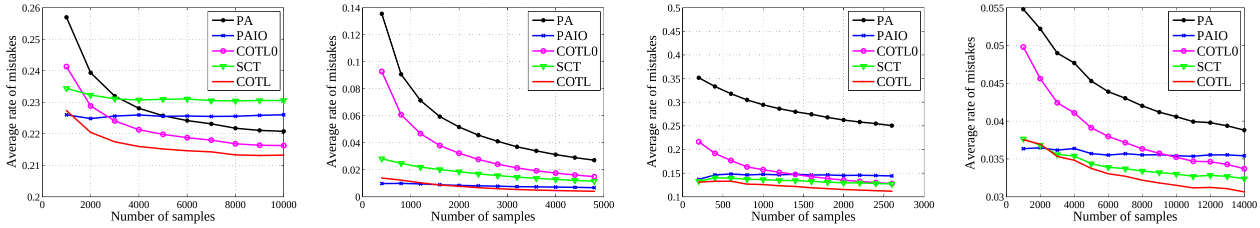
Proof of Lemma 1

Proof. Let $\Delta_t = \frac{\gamma_1}{2}\left(\|f_t^{(1)} - g^{(1)}\|^2 - \|f_{t+1}^{(1)} - g^{(1)}\|^2\right) + \frac{\gamma_2}{2}\left(\|f_t^{(2)} - g^{(2)}\|^2 - \|f_{t+1}^{(2)} - g^{(2)}\|^2\right)$, then

$$\begin{aligned}
\sum_{t=1}^{T}\Delta_t &= \sum_{t=1}^{T}\Big\{\frac{\gamma_1}{2}\left(\|f_t^{(1)} - g^{(1)}\|^2 - \|f_{t+1}^{(1)} - g^{(1)}\|^2\right)\\
&\quad + \frac{\gamma_2}{2}\left(\|f_t^{(2)} - g^{(2)}\|^2 - \|f_{t+1}^{(2)} - g^{(2)}\|^2\right)\Big\}\\
&= \frac{\gamma_1}{2}\left(\|h - g^{(1)}\|^2 - \|f_{T+1}^{(1)} - g^{(1)}\|^2\right)\\
&\quad + \frac{\gamma_2}{2}\left(\|f_1^{(2)} - g^{(2)}\|^2 - \|f_{T+1}^{(2)} - g^{(2)}\|^2\right)\\
&\leq \frac{\gamma_1}{2}\left(\|h - g^{(1)}\|^2\right) + \frac{\gamma_2}{2}\left(\|g^{(2)}\|^2\right)
\end{aligned}$$

Second, when $\ell_t = 0$, $f_{t+1}^{(i)} = f_t^{(i)}$ for $i = 1, 2$, it is clear $\Delta_t = 0$; when $\ell_t > 0$, $f_{t+1}^{(i)} = f_t^{(i)} + \frac{\tau_t}{2\gamma_i}y_{2_t}\kappa_i(x_{2_t}^{(i)}, \cdot)$, we compute Δ_t as:

$$\begin{aligned}
\Delta_t &= \frac{\gamma_1}{2}\left(\|f_t^{(1)} - g^{(1)}\|^2 - \|f_{t+1}^{(1)} - g^{(1)}\|^2\right)\\
&\quad + \frac{\gamma_2}{2}\left(\|f_t^{(2)} - g^{(2)}\|^2 - \|f_{t+1}^{(2)} - g^{(2)}\|^2\right)\\
&= \tau_t\Big\{ - \frac{y_{2_t}}{2}\left(f_t^{(1)}(x_{2_t}^{(1)}) + f_t^{(2)}(x_{2_t}^{(2)})\right)\\
&\quad + \frac{y_{2_t}}{2}\Big(g^{(1)}(x_{2_t}^{(1)}) + g^{(2)}(x_{2_t}^{(2)})\Big) - (\frac{k_t^1}{8\gamma_1} + \frac{k_t^2}{8\gamma_2})\tau_t\Big\} \qquad (10)
\end{aligned}$$



(a) *a7a* (b) mushrooms (c) spambase (d) w7a

Figure 4. Experimental results of average mistake rates on the heterogenous OTL tasks.

We also have $\ell_t = 1 - y_{2_t}\left(\frac{1}{2}\big(f_t^{(1)}(x_{2_t}^{(1)}) + f_t^{(2)}(x_{2_t}^{(2)})\big)\right)$ since $\ell_t > 0$. This is equivalent to the following:

$$\frac{y_{2_t}}{2}\big(f_t^{(1)}(x_{2_t}^{(1)}) + f_t^{(2)}(x_{2_t}^{(2)})\big) = 1 - \ell_t.$$

In addition,

$$\begin{aligned}\ell(g^{(1)}, g^{(2)}; t) &= \left[1 - y_{2_t}\frac{1}{2}\left(g^{(1)}(x_{2_t}^{(1)}) + g^{(2)}(x_{2_t}^{(2)})\right)\right]_+ \\ &\geq 1 - y_{2_t}\frac{1}{2}\left(g^{(1)}(x_{2_t}^{(1)}) + g^{(2)}(x_{2_t}^{(2)})\right),\end{aligned}$$

we thus have

$$\frac{y_{2_t}}{2}\left(g^{(1)}(x_{2_t}^{(1)}) + g^{(2)}(x_{2_t}^{(2)})\right) \geq 1 - \ell(g^{(1)}, g^{(2)}; (x_{2_t}, y_{2_t})).$$

Combining these two facts and inequality (10), we thus have the following result:

$$\begin{aligned}\Delta_t &\geq \tau_t\Big(-(1-\ell_t) + 1 - \ell(g^{(1)}, g^{(2)}; (x_{2_t}, y_{2_t})) - (\frac{k_t^1}{8\gamma_1} + \frac{k_t^2}{8\gamma_2})\tau_t\Big) \\ &= \tau_t\Big(\ell_t - \ell(g^{(1)}, g^{(2)}; (x_{2_t}, y_{2_t})) - (\frac{k_t^1}{8\gamma_1} + \frac{k_t^2}{8\gamma_2})\tau_t\Big)\end{aligned}$$

Hence, we have the following conclusion:

$$\begin{aligned}&\sum_{t=1}^{T}\tau_t\left(\ell_t - \ell(g^{(1)}, g^{(2)}; (x_{2_t}, y_{2_t})) - (\frac{k_t^1}{8\gamma_1} + \frac{k_t^2}{8\gamma_2})\tau_t\right) \\ &\leq \frac{\gamma_1}{2}\|h - g^{(1)}\|^2 + \frac{\gamma_2}{2}\|g^{(2)}\|^2\end{aligned}$$

□

References

Argyriou, Andreas, Maurer, Andreas, and Pontil, Massimiliano. An algorithm for transfer learning in a heterogeneous environment. In *Euro. Conf. Mach. Learn. and Knowledge Discovery in Databases*, pp. 71–85, Antwerp, Belgium, 2008.

Arnold, Andrew, Nallapati, Ramesh, and Cohen, William W. A comparative study of methods for transductive transfer learning. In *Proc. 7th IEEE Int'l Conf. on Data Mining Workshops*, pp. 77–82, Washington, DC, USA, 2007.

Cesa-Bianchi, Nicolo and Lugosi, Gabor. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006. ISBN 0521841089.

Crammer, Koby, Dekel, Ofer, Keshet, Joseph, Shalev-Shwartz, Shai, and Singer, Yoram. Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585, 2006.

Dai, Wenyuan, Yang, Qiang, Xue, Gui-Rong, and Yu, Yong. Self-taught clustering. In *Proc. 25th Int'l Conf. on Machine Learning (ICML2008)*, pp. 200–207, Helsinki, Finland, 2008.

DaumáIII, H. and Marcu, D. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126, 2006.

Dekel, Ofer, Long, Philip M., and Singer, Yoram. Online learning of multiple tasks with a shared loss. *J. Mach. Learn. Res.*, 8:2233–2264, 2007.

Freund, Yoav and Schapire, Robert E. Large margin classification using the perceptron algorithm. *Mach. Learn.*, 37(3):277–296, 1999.

Li, Yi and Long, Philip M. The relaxed online maximum margin algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 498–504, 1999.

Pan, Sinno Jialin and Yang, Qiang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2009.

Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958.

Shalev-Shwartz, Shai. Online learning:theory, algorithms, and applications. In *Ph.D thesis*, 2007.

Yang, Haiqin, Xu, Zenglin, King, Irwin, and Lyu, Michael. Online learning for group lasso. In *27th Int'l Conf. on Machine Learning (ICML2010)*, Haifa, Israel, 2010.

Zhao, Peilin, Hoi, Steven C. H., and Jin, Rong. Duol: A double updating approach for online learning. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2259–2267, 2009.