Online Heterogeneous Transfer by Hedge Ensemble of Offline and Online Decisions

Yuguang Yan, Qingyao Wu, Mingkui Tan, Michael K. Ng, Huaqing Min, and Ivor W. Tsang

***Abstract*— In this paper, we study the online heterogeneous transfer (OHT) learning problem, where the target data of interest arrive in an online manner, while the source data and auxiliary co-occurrence data are from offline sources and can be easily annotated. OHT is very challenging, since the feature spaces of the source and target domains are different. To address this, we propose a novel technique called *OHT by hedge ensemble* by exploiting both offline knowledge and online knowledge of different domains. To this end, we build an offline decision function based on a heterogeneous similarity that is constructed using labeled source data and unlabeled auxiliary co-occurrence data. After that, an online decision function is learned from the target data. Last, we employ a hedge weighting strategy to combine the offline and online decision functions to exploit knowledge from the source and target domains of different feature spaces. We also provide a theoretical analysis regarding the mistake bounds of the proposed approach. Comprehensive experiments on three real-world data sets demonstrate the effectiveness of the proposed technique.**

***Index Terms*— Co-occurrence data, hedge weighting, heterogeneous transfer learning (HTL), online learning.**

I. Introduction

TRANSFER learning (TL) seeks to improve the learning performance in a target domain by leveraging knowledge from a source domain with a different data distribution or feature space [1]–[3]. TL has been extensively explored in situations where training data in the target domain of interest are limited or too expensive to collect. To this end, in the past decade, a number of TL methods have been proposed [4]–[7]. Most of the methods are focused on homogeneous settings, in which the source and target data share a common feature space [8]–[10].

Recently, a new TL scheme, called heterogeneous TL (HTL), has attracted great attention [11]–[14]. Unlike in the homogeneous setting, the source and target data in HTL come from two different feature spaces. HTL has been applied in many applications in computer vision and machine learning [15]–[17]. For example, for the image classification task, it is common to have a limited number of annotated images but with much labeled text information as auxiliary source knowledge. The image and text data are represented by different kinds of features. Therefore, determining how to exploit the labeled text data to boost the prediction performance over images is very challenging.

Fortunately, as will be shown in this paper, it is possible to collect some text data related to images as co-occurrence data, such as image captions or text from documents containing images. In this case, the knowledge from text data could be appropriately transferred, which will help improve the performance of image classification [11]. It is worth mentioning that, in contrast to the expensive cost of image labeling, the unlabeled text-image co-occurrence data can be easily collected from many sources. For example, the website Flickr [1] contains a tremendous number of images with tags, and some social networks include a large number of pictures with text comments posted by users.

While the effectiveness of HTL has been demonstrated by many works [11], [14], [18]–[20], most existing studies are focused on the offline/batch learning problem by assuming that all the training instances from the target domain are accessible in advance. However, this assumption may not conform to real-world applications, where the target instances arrive in an online manner. For example, users may chance upon interesting pictures, and then post and share them occasionally. As a result, the social-network platform receives and publishes these images in an online/sequential manner. Motivated by this, we focus on the online heterogeneous transfer (OHT) learning problem, where the target data of interest arrive in an online manner, while the source data are collected offline and annotated, with the co-occurrence data given as auxiliary information.

There are two main challenges for OHT. First, the feature spaces of the source and target domains are completely

Manuscript received May 20, 2016; revised November 15, 2016, March 7, 2017, and August 31, 2017; accepted August 31, 2017. This work was supported in part by the Guangzhou Key Laboratory of Robotics and Intelligent Software under Grant 15180007, in part by the National Natural Science Foundation of China under Grant 61502177 and Grant 61602185, in part by the Fundamental Research Funds for the Central Universities under Grant D2172500 and Grant D2172480, in part by the CCF-Tencent Open Research Fund, in part by the Guangdong Provincial Scientific and Technological Fund under Grant 2017B090901008 and Grant 2017A010101011, in part by HKRGC GRF under Grant 12302715, Grant 12306616, and Grant 12200317, in part by HKBU under Grant RC-ICRS/16-17/03, in part by the ARC Future Fellowship under Grant FT130100746, and in part by the ARC under Grant LP150100671. *(Corresponding authors: Qingyao Wu; Mingkui Tan.)*

Y. Yan, Q. Wu, M. Tan, and H. Min are with the School of Software Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: yan.yuguang@mail.scut.edu.cn; qyw@scut.edu.cn; mingkuitan@scut.edu.cn; hqmin@scut.edu.cn).

M. K. Ng is with the Mathematics Department, Hong Kong Baptist University, Hong Kong (e-mail: mng@math.hkbu.edu.hk).

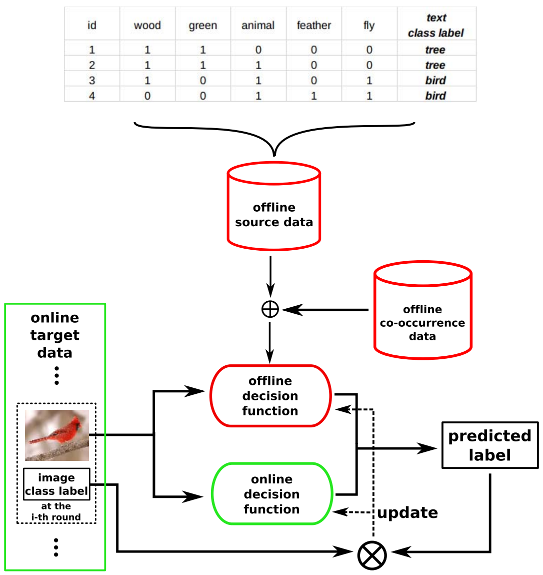
I. W. Tsang is with the Center for Quantum Computation and Intelligent Systems, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: ivor.tsang@uts.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2751102

[1] http://www.flickr.com

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



id	wood	green	animal	feather	fly	*text class label*
1	1	1	0	0	0	*tree*
2	1	1	1	0	0	*tree*
3	1	0	1	0	1	*bird*
4	0	0	1	1	1	*bird*

Fig. 1. Overall schematic of the proposed OHTHE scheme that exploits knowledge from source and target domains of different feature spaces, where the text and image data are considered as the source and target domains, respectively. The ⊕ marker means that we combine the source and co-occurrence data to obtain the offline decision function, and the ⊗ marker means that we measure the difference between the true label and the predicted label to update the classifier. At each round, the classifier receives an instance from the target domain and predicts its label by exploiting both offline and online decision functions. After that, the classifier receives the true label to evaluate whether the prediction is correct, and updates itself according to the suffered loss.

different; thus, we cannot directly use the source data to boost the learning performance in the target domain. Second, since the labeled training data of the target domain are limited, it is difficult to build a precise correspondence map to bridge the source and target domains.

To address these challenges, we propose a novel technique called *OHT by hedge ensemble* (OHTHE) by exploiting both source knowledge and target knowledge to boost the learning performance in the target domain. Taking an image classification task for example, the general scheme of OHTHE is shown in Fig. 1, where the text information is adopted to boost the performance.

In this paper, we make the following contributions.

1) To connect the source and target domains, we build an offline decision function based on a heterogeneous similarity that is constructed using both the labeled source data and the unlabeled co-occurrence data.
2) We employ a hedge weighting strategy to exploit both offline and online decision functions to boost the learning performance. Specifically, OHTHE adaptively adjusts the weights of two kinds of decision functions according to the differences between the true labels and the decision values.
3) We justify the proposed technique by providing theoretical mistake bounds of the proposed algorithms and conducting comprehensive empirical studies on three real-world data sets.

The rest of this paper is organized as follows. First, we discuss related works in Section II. After that, we present the proposed OHTHE technique in Section III and conduct the theoretical analysis in Section IV. We present the experimental study in Section V and conclude this paper in Section VI.

II. Related Works

A. Transfer Learning

Pan and Yang [1] categorized TL into three classes: inductive transfer [4]–[6], [21], transductive transfer [22], and unsupervised transfer [23], [24]. Shao *et al.* [3] summarized that three kinds of knowledge are useful for transfer: 1) source domain features; 2) source domain features and the corresponding labels; and 3) parameters of the source domain models. Based on this, the knowledge transferred in our proposed technique consists of source domain features and the corresponding labels.

TL can also be divided into two categories according to the feature spaces of the source and target domains, namely, homogeneous transfer and heterogeneous transfer. Homogeneous TL addresses the situation where the source and target data are in the same feature space, and it has been widely applied in many real-world applications, such as text mining [6], [7], [21], image classification [4], [25], and face recognition [26]. References [25] and [26] used a boosting strategy to adjust the weights of source and target data. References [25] and [26] studied feature learning models involving low-rank constraints. Pan *et al.* [8] proposed transfer component analysis, a feature extraction method for TL. Cheng and Pan [9] addressed a semisupervised setting by learning on manifolds. Li *et al.* [10] reweighted the predictions of a source classifier for target test data.

For HTL, Shi *et al.* [27] employed spectral transformation to map source and target data into a common subspace. Wei and Pal [28] applied restricted Boltzmann machine to perform HTL tasks. Pan and Yang [29] proposed to leverage binary ratings (e.g., like or dislike) in a source domain to alleviate the sparsity issue in target numerical ratings. A deep learning technique was introduced into HTL in [30]. Zhou *et al.* [31] proposed a method to perform HTL for multiclass problems based on the compressed sensing theory. Reference [32] studied HTL in the framework of reinforcement learning. Niu *et al.* [14] leveraged heterogeneous Web sources to assist in action and event recognition in videos.

Co-occurrence data have been used to address OHT problems. Dai *et al.* [18] constructed a translator to connect two different feature spaces by exploiting co-occurrence data. Yang *et al.* [33] leveraged text data for image clustering. Zhu *et al.* [15], Wang *et al.* [19], and Qi *et al.* [34] utilized text-image co-occurrence data to perform image classification tasks. Ng *et al.* [16], Wu *et al.* [17], and Tan *et al.* [20] used a transition probability matrix to address a classification task in a target domain. Recently, Yang *et al.* [11] studied a scheme to evaluate the relatedness among given source domains through transferred weights, which are learned from co-occurrence data. Yang *et al.* [12] developed a robust matrix factorization model for heterogeneous transfer from text data to image data.

Pan and Yang [1] provided a comprehensive survey of TL. For more recent advances in visual TL, please refer to [2] and [3].

B. Online Transfer Learning

Online learning has been extensively studied in the machine learning community [35], [36]. Perceptron [37] simply updates a linear classifier when a new instance is classified incorrectly. Freund and Schapire [38] proposed a hedge strategy to combine several predicted results given by multiple decisions. In [39] and [40], the maximum margin was introduced into online learning. Zinkevich [41] and Shalev-Shwartz *et al.* [42], [43] updated classifiers by gradient-based algorithms, and then projected the results into constrained spaces. Recently, the second-order information was considered in [44]–[46], and a confidence-weighted strategy was proposed to update the classifiers.

In the literature, a few approaches have been proposed to address online TL problems. In [47], online TL was studied in the multiarmed bandit framework. In [48] and [49], online TL with online homogeneous source data was studied. Zhao *et al.* [50], [51] used some offline data to assist in an online task in a target domain. They used the ensemble strategy to tackle online homogeneous TL problems, and employed a multiview approach to handling OHT learning problems. References [50] and [51] considered OHT under the assumption that the feature space of the source domain is a subset of that of the target domain.

Last, we highlight the difference between OHT and multiview learning [52]. For multiview learning, each instance is associated with multiple views (sets) of features, while in OHT, a target instance is represented by one set of features and a source instance is represented by another set of features.

III. Proposed Method

A. Problem Definition

In the OHT learning problem, the target instances $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n} \in \mathcal{X} \times \mathcal{Y}$ arrive in an online manner, while the labeled source data $\{(\mathbf{x}_l^s, y_l^s)\}_{l=1}^{n_s} \in \mathcal{X}_s \times \mathcal{Y}_s$ come from an offline source domain, where n and n_s refer to the numbers of target and source data, respectively. Here, $\mathcal{Y}_s = \mathcal{Y} = \{+1, -1\}$ denotes the common label space of the source and target domains. Recall that in OHT, the source feature space $\mathcal{X}_s = \mathbb{R}^{d_s}$ is different from the target feature space $\mathcal{X} = \mathbb{R}^{d}$. In particular, the dimensions of the source and target data are different, i.e., $d_s \neq d$.

1) Co-Occurrence Data: The unlabeled co-occurrence data $\{(\mathbf{u}_j, \mathbf{v}_j)\}_{j=1}^{n_c} \in \mathcal{X}_c$ are from offline sources, where $\mathbf{u}_j \in \mathcal{X}_s$ and $\mathbf{v}_j \in \mathcal{X}$. For example, in Flickr, users posted pictures and added some tags to describe them. As a result, each image is associated with some tags, which can be used as co-occurrence data to connect text and image data. Fig. 2 shows the text-image co-occurrence data that are collected from Flickr. In this example, $\mathbf{u}_j$ represents the text part of the jth co-occurrence instance, while $\mathbf{v}_j$ represents the image part of the jth co-occurrence instance.

B. General Scheme

The objective of OHT is to learn a classifier $f(\mathbf{x})$ to predict the label of an unseen target instance $\mathbf{x}$, which arrives in an online fashion. Directly constructing an online decision function $h(\mathbf{x})$ on the target data could be the most straightforward approach to learning $f(\mathbf{x})$, i.e., $f(\mathbf{x}) = \text{sign}\,(h(\mathbf{x}))$. However, this approach is limited when the number of training samples is limited, since the performance of an online decision function is highly dependent on the number of training instances.

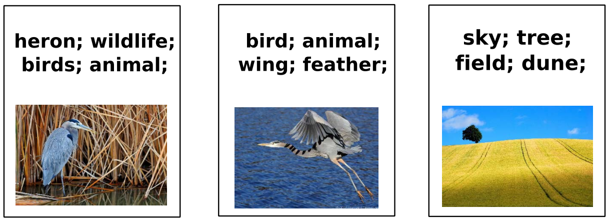


Fig. 2. Examples of text-image co-occurrence data collected from Flickr.

On the other hand, in the OHT problem, we have sufficient source and unlabeled co-occurrence data, which may provide additional information for the prediction for the target data. As a result, by taking advantage of the source and co-occurrence data, we may build an offline source decision function $h^s(\mathbf{x})$ that reflects the knowledge from both source and co-occurrence data, and then use it to boost the prediction performance on the target data. Nevertheless, even though $h^s(\mathbf{x})$ is given, determining how to exploit the knowledge of $h^s(\mathbf{x})$ remains a critical issue.

Without loss of generality, we suppose that the offline source decision function $h^s(\mathbf{x})$ is given and $h(\mathbf{x})$ is updated in an online fashion. To boost the performance on the target data, we propose to combine the offline decision function $h^s(\mathbf{x})$ and the online decision function $h(\mathbf{x})$. Mathematically, we apply a convex combination of $h^s(\mathbf{x})$ and $h(\mathbf{x})$ to build the final ensemble classifier $f(\mathbf{x})$, which predicts the label for $\mathbf{x}_i$ as follows:

$$f(\mathbf{x}_i) = \text{sign}\left(\theta_i^s \phi(h^s(\mathbf{x}_i)) + \theta_i \phi(h(\mathbf{x}_i)) - \frac{1}{2}\right) \tag{1}$$

where we constrain the mapping function $\phi(\cdot) \in [0, 1]$. θ_i^s and θ_i are the weights with respect to the two kinds of decision functions, where $\theta_i^s, \theta_i \in [0, 1]$, and $\theta_i^s + \theta_i = 1$. As a result, $\left(\theta_i^s \phi(h^s(\mathbf{x}_i)) + \theta_i \phi(h\ (\mathbf{x}_i))\right) \in [0, 1]$, and the constant $1/2$ can be considered as a threshold.

Note that the weights θ_i^s and θ_i are also updated online. Here, we employ the **Hedge**(β) strategy [38] to update them dynamically. Specifically, at the ith round, the **Hedge**(β) strategy predicts the label for $\mathbf{x}_i$ by the predictor function (1) and then updates the two weights by the following rules:

$$\theta_{i+1}^s = \theta_i^s \beta^{\psi\ (y_i h^s(\mathbf{x}_i))}, \quad \theta_{i+1} = \theta_i \beta^{\psi\ (y_i h(\mathbf{x}_i))} \tag{2}$$

where $\beta \in\ (0, 1)$ is a decay factor, and ψ is some loss function to determine the degree of decay. Equation (2) implies that a larger loss incurred by ψ will result in a larger degree of decay. In other words, the better decision function will have a relatively greater contribution in the combination. Once θ_i^s and θ_i are updated, we perform normalization, such that $\theta_{i+1}^s + \theta_{i+1} = 1$ by

$$\theta_{i+1}^s := \frac{\theta_{i+1}^s}{\theta_{i+1}^s + \theta_{i+1}}, \quad \theta_{i+1} := \frac{\theta_{i+1}}{\theta_{i+1}^s + \theta_{i+1}}. \tag{3}$$

The mappings ϕ and ψ play important roles in the ensemble. For convenience, we leave the detailed discussions of them later.

Suppose the offline source decision function $h^s(\mathbf{x})$ is given and $h(\mathbf{x})$ is a linear function, i.e., $h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$. The scheme of the proposed *OHTHE* is presented in Algorithm 1.

Algorithm 1 OHTHE

```
Input: The heterogeneous source data $\{\mathbf{x}_i^s, y_i^s\}_{i=1}^{n_s}$, co-occurrence data $\{(\mathbf{u}_j, \mathbf{v}_j)\}_{j=1}^{n_c}$, the regularization parameter $c > 0$, the decay factor $\beta \in (0, 1)$.
Initialize: $\mathbf{w}_1 = \mathbf{0}$, $\theta_1^s \in (0, 1)$ and $\theta_1 \in (0, 1)$ with $\theta_1^s + \theta_1 = 1$.
1: for $i = 1$ to $n$ do
2:    Receive a new instance $\mathbf{x}_i \in \mathcal{X}$.
3:    Make the prediction:
          Compute $h^s(\mathbf{x}_i)$ from Eq. (9).
          Compute $h(\mathbf{x}_i) = \mathbf{w}_i^\top \mathbf{x}_i$.
          Calculate prediction according to
          $\hat{y}_i = \text{sign}\left(\theta_i^s \phi(h^s(\mathbf{x}_i)) + \theta_i \phi(h(\mathbf{x}_i)) - \frac{1}{2}\right)$.
4:    Obtain the true label $y_i$.
5:    Update the online target decision function $h(\mathbf{x}_i)$:
          Compute $\ell_i^* = \max\{0, 1 - y_i\ (\mathbf{w}_i^\top \mathbf{x}_i)\}$.
          Compute $\tau_i = \min\{c, \frac{\ell_i^*}{\|\mathbf{x}_i\|^2}\}$.
          Compute $\mathbf{w}_{i+1}$ from Eq. (5).
6:    Update the weights:
          Compute $\theta_{i+1}^s$ and $\theta_{i+1}$ from Eq. (2).
          Perform normalization such that $\theta_{i+1}^s + \theta_{i+1} = 1$ by Eq. (3).
7: end for
```

Note that in Step 5 of Algorithm 1, we need to update the online decision function $h(\mathbf{x})$. Here, we apply the PA scheme to update $\mathbf{w}_i$ by

$$\begin{aligned} \mathbf{w}_{i+1} = \ & \arg\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2}||\mathbf{w} - \mathbf{w}_i||^2 + c\xi \\ & \text{s.t. } \ell^*(\mathbf{x}_i, y_i; \mathbf{w}) \le \xi \quad \text{and} \quad \xi \ge 0 \end{aligned} \tag{4}$$

where $c > 0$ is a regularization parameter, $\ell^*(\mathbf{x}, y; \mathbf{w}) = \max\{1 - y\ (\mathbf{w}^\top \mathbf{x}), 0\}$ is the hinge loss, and $||\cdot||$ is the Euclidean norm. Problem (4) requires $\mathbf{w}_{i+1}$ to correctly classify the current instance $\mathbf{x}_i$ with a sufficiently large margin and to stay as close as possible to $\mathbf{w}_i$. This problem has a closed-form solution

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \tau_i y_i \mathbf{x}_i \tag{5}$$

where $\tau_i = \min\left\{c, \frac{\ell^*(\mathbf{x}_i, y_i; \mathbf{w}_i)}{||\mathbf{x}_i||^2}\right\}$.

C. Learning the Source Decision Function

Now, we are ready to detail the construction of the source decision function $h^s(\mathbf{x})$, which should reflect the knowledge in the source and co-occurrence data. In order to leverage the label information of the heterogeneous source data, we use the labels of the k nearest neighbors of a target instance from the source data to assist in the prediction. To this end, we need a heterogeneous similarity that measures the relationship between a target instance and a heterogeneous source instance. The principal issue is that the standard similarity measures the relationship between two homogeneous instances; thus, it cannot be used directly to measure the relationship between two heterogeneous instances $\mathbf{x}_i$ and $\mathbf{x}_l^s$. To address this, we propose to use the co-occurrence data $\{(\mathbf{u}_j, \mathbf{v}_j)\}_{j=1}^{n_c}$ to connect $\mathbf{x}_i$ and $\mathbf{x}_l^s$ so that the heterogeneous similarity between $\mathbf{x}_i$ and $\mathbf{x}_l^s$ can be calculated. The main idea is to calculate two kinds of similarities between homogeneous instances: one is between $\mathbf{x}_i$ and $\mathbf{v}_j$, and the other one is between $\mathbf{u}_j$ and $\mathbf{x}_l^s$. After that, the heterogeneous similarity can be obtained by combining these two kinds of similarities. In the following, we discuss the calculation method in detail.

Note that the target data arrive in an online manner. When a target instance $\mathbf{x}_i$ arrives, we measure the similarity between $\mathbf{x}_i$ and $\mathbf{v}_j$, which is denoted by $\rho(\mathbf{x}_i, \mathbf{v}_j)$, using the Pearson correlation

$$\rho(\mathbf{x}_i, \mathbf{v}_j) = \frac{\sum_{p=1}^{d}(x_{i,p} - \bar{x}_i)(v_{j,p} - \bar{v}_j)}{\epsilon + \sqrt{\sum_{p=1}^{d}(x_{i,p} - \bar{x}_i)^2}\sqrt{\sum_{p=1}^{d}(v_{j,p} - \bar{v}_j)^2}} \tag{6}$$

where ϵ is a very small constant (e.g., $\epsilon = 1\text{e}{-10}$), which is used to avoid numerical issues. $x_{i,p}$ is the pth element of the vector $\mathbf{x}_i$, $\bar{x}_i = \sum_{p=1}^{d} x_{i,p}$; and analogously for $v_{j,p}$ and $\bar{v}_j$. Similarly, the similarity $\varrho(\mathbf{u}_j, \mathbf{x}_l^s)$ between $\mathbf{u}_j$ and $\mathbf{x}_l^s$ is calculated as

$$\varrho(\mathbf{u}_j, \mathbf{x}_l^s) = \frac{\sum_{p=1}^{d_s}(u_{j,p} - \bar{u}_j)(x_{l,p}^s - \bar{x}_l^s)}{\epsilon + \sqrt{\sum_{p=1}^{d_s}(u_{j,p} - \bar{u}_j)^2}\sqrt{\sum_{p=1}^{d_s}(x_{l,p}^s - \bar{x}_l^s)^2}}. \tag{7}$$

Given $\rho(\mathbf{x}_i, \mathbf{v}_j)$ and $\varrho(\mathbf{u}_j, \mathbf{x}_l^s)$, the similarity $\chi(\mathbf{x}_i, \mathbf{x}_l^s)$ between $\mathbf{x}_i$ and $\mathbf{x}_l^s$ can be obtained by using all the co-occurrence instances as follows:

$$\chi(\mathbf{x}_i, \mathbf{x}_l^s) = \sum_{j=1}^{n_c} \rho(\mathbf{x}_i, \mathbf{v}_j)\varrho(\mathbf{u}_j, \mathbf{x}_l^s). \tag{8}$$

Consequently, based on the heterogeneous similarity $\chi(\mathbf{x}_i, \mathbf{x}_l^s)$, the offline decision function is able to make a prediction for $\mathbf{x}_i$ by using the label information of the source data. Specifically, the offline decision function finds the k nearest neighbors of $\mathbf{x}_i$ from the source data and then computes the weighted sum of the labels of these neighbors as follows:

$$h^s(\mathbf{x}_i) = \sum_{\tau \in N} y_\tau^s \chi(\mathbf{x}_i, \mathbf{x}_\tau^s) \Big/ \sum_{\tau \in N} \chi(\mathbf{x}_i, \mathbf{x}_\tau^s) \tag{9}$$

where the set N includes the indices of $\mathbf{x}_i$ values k nearest neighbors that are found from the source instances.

D. Design of ϕ and ψ

Now, we discuss the mapping functions ϕ and ψ used in (1) and (2), where ϕ maps a decision value into the range [0, 1]

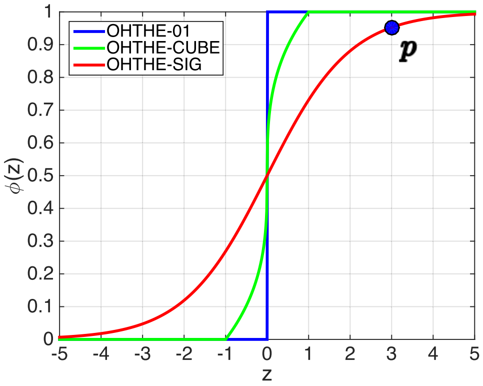


Fig. 3. Mapping functions $\phi(z) \in [0, 1]$, where z is the decision value. The point p in the figure is attained with a large z value.

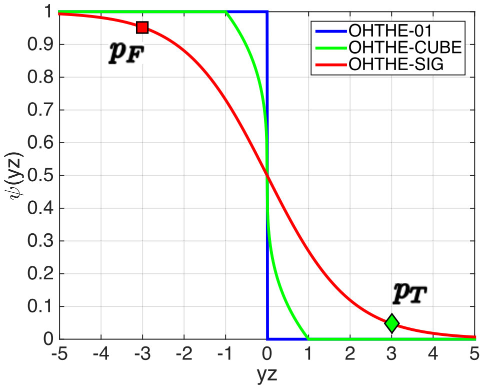


Fig. 4. Loss functions $\psi(yz) \in [0, 1]$, where z is the decision value, and y is the true label. For the point p in Fig. 3, if $y = 1$, we obtain a small loss (see the point p_T); otherwise, if $y = -1$, we obtain a large loss (see the point p_F).

and ψ generates a loss that determines the degree of decay. For simplicity, let z be a decision value given by $h(\mathbf{x})$ or $h^s(\mathbf{x})$, whose absolute value measures the confidence we have in this prediction. Rationally, the intuition is that the loss $\psi(yz)$ must highly depend on the correctness of the prediction and the confidence we have. Consider a situation where we get a decision value z with a large absolute value. If our prediction is correct, we should obtain a small loss. In contrast, if our prediction is incorrect, we should suffer a large loss. This intuition motivates the design of functions ϕ and ψ and is shown in Figs. 3 and 4.

Meanwhile, in order to obtain theoretical guarantees, which will be given in Section IV, we seek pairs of ϕ and ψ that satisfy the conditions $\phi(z) = \psi(-z)$ and $\phi(z) + \psi(z) = 1$. Since $y \in \{+1, -1\}$, we have $\psi(yz) = \psi(z)$ or $\psi(-z)$. In the following, we present several examples, each of which represents a version of OHTHE.

1) OHTHE-01

$$\phi(z) = \begin{cases} 1, & \text{if } z > 0 \\ 0, & \text{if } z \le 0 \end{cases} \tag{10}$$

$$\psi(yz) = \begin{cases} 0, & \text{if } yz > 0 \\ 1, & \text{if } yz \le 0. \end{cases} \tag{11}$$

2) OHTHE-CUBE

$$\phi(z) = \max\left\{\min\left\{\frac{1}{2}z^{1/3} + \frac{1}{2}, 1\right\}, 0\right\} \tag{12}$$

$$\psi(yz) = \max\left\{\min\left\{\frac{1}{2}(-yz)^{1/3} + \frac{1}{2}, 1\right\}, 0\right\}. \tag{13}$$

3) OHTHE-SIG

$$\phi(z) = \frac{1}{1 + \exp\{-z\}} \tag{14}$$

$$\psi(yz) = \frac{1}{1 + \exp\{yz\}}. \tag{15}$$

E. Computational Complexity

We first study the complexity of the offline computation, and then analyze the complexity of each round of online learning. Since the source and co-occurrence data come from offline sources, we compute the similarity between each source instance and each co-occurrence instance in (7) before the online learning task begins. For each co-occurrence instance, the complexity of (7) is $O(d_s)$, where d_s is the dimension of the source instance. Therefore, the complexity of the offline computation is $O(n_s n_c d_s)$, which involves n_s source instances and n_c co-occurrence instances.

For the computation at each round of online learning, we first analyze the complexity of Step 3 in Algorithm 1. Equation (6) computes the similarity between the current target instance and each co-occurrence instance, which has a total cost of $O(n_c d)$, where d is the dimension of the target instance. For n_s source instances, the complexity of (8) is $O(n_s n_c)$. Finding k nearest neighbors in (9) has a cost of $O(n_s \log n_s)$. By adding the complexity $O(d)$ of prediction in (1), the complexity of Step 3 is $O(n_c d + n_s n_c + n_s \log n_s + d)$. Step 5 involves addition and inner product calculations of vectors; thus, it has a cost of $O(d)$. Step 6 has $O(1)$ complexity. Finally, the total complexity for n target instances is $O(n_s n_c d_s + n\ (n_c d + n_s n_c + n_s \log n_s + d))$.

IV. Theoretical Analysis

In this section, we study the theoretical bounds of the proposed OHTHE algorithms. We first establish Proposition 1 [38].

Proposition 1: Let $\ell_i^s = \psi(y_i h^s(\mathbf{x}_i))$, $\ell_i = \psi(y_i h(\mathbf{x}_i))$, and β be the decay factor of the weights. θ_i^s and θ_i are the normalized weights of two decision functions with $\theta_i^s + \theta_i = 1$. When $\ell_i^s \in [0, 1]$, $\ell_i \in [0, 1]$, and $\beta \in (0, 1)$, for any sequence of loss vectors $\{(\ell_i^s, \ell_i) | i = 1, 2, \ldots n\}$, we have

$$\sum_{i=1}^{n} (\theta_i^s \ell_i^s + \theta_i \ell_i) \le \frac{1}{1-\beta} \min\{\Delta^s, \Delta\} \tag{16}$$

where $\Delta^s = \ln(1/\theta_1^s) + (\ln\ (1/\beta)) \sum_{i=1}^{n} \ell_i^s$ and $\Delta = \ln(1/\theta_1) + (\ln\ (1/\beta)) \sum_{i=1}^{n} \ell_i$.

Remark: Proposition 1 states that the entire loss, which is the sum of the losses at all n rounds, is not much larger than t loss suffered by the better of the two decision functions. On the right-hand side of inequality (16), $\ln(1/\theta_1^s)$ and $\ln(1/\theta_1)$ measure the prior confidences that we have in the two decision functions. If there is no preference, we can simply set $\theta_1^s = \theta_1 = (1/2)$. At this time, the upper bound only depends on the sum of the separate losses of the two decision functions. On the other hand, if we have some prior information on which decision function will perform better, we can change $\ln(1/\theta_1^s)$ and $\ln(1/\theta_1)$ to get a tighter bound. For instance, if we believe that the online target decision function will achieve a lower loss, which means that $\sum_{i=1}^n \ell_i < \sum_{i=1}^n \ell_i^s$, we can set $\theta_1 > \theta_1^s$. If our guess is correct, we can obtain a better bound than the one we obtained by setting $\theta_1 = \theta_1^s$. In the following, we analyze the theoretical results that are obtained by setting $\theta_1^s = \theta_1 = (1/2)$.

According to Proposition 1, the mistake bound of OHTHE is given by Theorem 1.

Theorem 1: Given $\theta_1 = \theta_1^s = (1/2)$, let M be the number of the mistakes made by a version of OHTHE after receiving a sequence of n instances. Then, we have

$$M \le \frac{2}{1-\beta}\min\{\mathit{\Delta}^s, \mathit{\Delta}\} \tag{17}$$

where $\mathit{\Delta}^s = \ln 2 + (\ln\ (1/\beta))\sum_{i=1}^n \ell_i^s$ and $\mathit{\Delta} = \ln 2 + (\ln\ (1/\beta))\sum_{i=1}^n \ell_i$.

Proof: Note that for all three pairs of $\phi(z)$ and $\psi(z)$ in Section III-D, we have

$$\phi(z) = \psi(-z) \tag{18}$$
$$\phi(z) + \psi(z) = 1. \tag{19}$$

When the ensemble classifier makes a mistake at the ith round, according to (1), we should have

$$y_i\hat{y}_i = y_i\left(\text{sign}\left(\theta_i^s\phi(h^s(\mathbf{x}_i)) + \theta_i\phi(h\ \ (\mathbf{x}_i)) - \frac{1}{2}\right)\right) < 0. \tag{20}$$

There are two possible values of the true label y_i: 1 and -1.

Case 1 *(*$y_i = 1$*):* We know that $\theta_i^s\phi(h^s(\mathbf{x}_i)) + \theta_i\phi(h\ \ (\mathbf{x}_i)) < (1/2)$. Therefore, we have

$$\begin{aligned}
&\theta_i^s\phi(h^s(\mathbf{x}_i)) + \theta_i\phi(h\ \ (\mathbf{x}_i))\\
&\quad = \theta_i^s\left(1 - \psi(h^s(\mathbf{x}_i))\right) + \theta_i\left(1 - \psi(h\ \ (\mathbf{x}_i))\right)\\
&\quad = \theta_i^s - \theta_i^s\psi(h^s(\mathbf{x}_i)) + \theta_i - \theta_i\psi(h\ \ (\mathbf{x}_i))\\
&\quad = 1 - \left(\theta_i^s\psi(h^s(\mathbf{x}_i)) + \theta_i\psi(h\ \ (\mathbf{x}_i))\right) < \frac{1}{2}.
\end{aligned}$$

By using $y_i = 1$, we obtain $(1/2) < \theta_i^s\psi(y_i h^s(\mathbf{x}_i)) + \theta_i\psi(y_i h\ \ (\mathbf{x}_i))$.

Case 2 *(*$y_i = -1$*):* We know that $\theta_i^s\phi(h^s(\mathbf{x}_i)) + \theta_i\phi(h\ \ (\mathbf{x}_i)) > (1/2)$. Thus, we obtain

$$\theta_i^s\phi(h^s(\mathbf{x}_i)) + \theta_i\phi(h\ \ (\mathbf{x}_i)) = \theta_i^s\psi(-h^s(\mathbf{x}_i)) + \theta_i\psi(-h\ \ (\mathbf{x}_i)) > \frac{1}{2}.$$

By using $y_i = -1$, we get $(1/2) < \theta_i^s\psi(y_i h^s(\mathbf{x}_i)) + \theta_i\psi(y_i h\ \ (\mathbf{x}_i))$. In summary, we always have

$$\frac{1}{2} < \theta_i^s\psi(y_i h^s(\mathbf{x}_i)) + \theta_i\psi(y_i h\ \ (\mathbf{x}_i)).$$

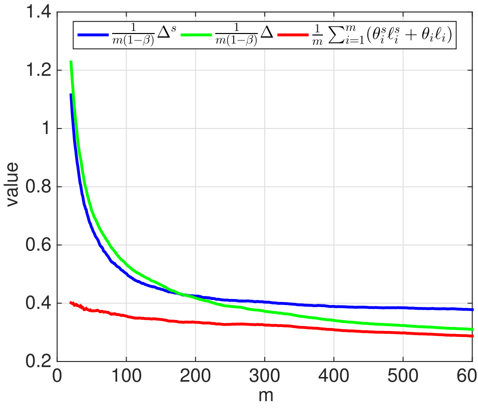


Fig. 5. Results related to the loss bound in Proposition 1, where m is the number of target instances that have been received. The blue and green lines represent the results related to the offline and online decision functions, respectively. The red line represents the result produced by OHTHE-SIG.

By adding the inequalities of all the mistakes, we obtain

$$\frac{1}{2}M < \sum_{i=1}^n\left(\theta_i^s\psi(y_i h^s(\mathbf{x}_i)) + \theta_i\psi(y_i h\ \ (\mathbf{x}_i))\right).$$

The theorem follows directly from Proposition 1 by multiplying both sides of the above inequality by 2. □

Next, we suggest a value of β based on the following lemma and theorem.

Lemma 1: Let $L = \sum_{i=1}^n \ell_i^s$ or $L = \sum_{i=1}^n \ell_i$. Suppose $L \le \tilde{L}$. When $\beta = (\sqrt{\tilde{L}}/(\sqrt{\tilde{L}} + \sqrt{2\ln 2}))$, we have

$$\frac{-L\ln\beta + \ln 2}{1-\beta} \le L + \sqrt{2\tilde{L}\ln 2} + \ln 2. \tag{21}$$

This lemma can be derived from [38, Lemma 4]. $\tilde{L}$ is an upper bound of the cumulative loss suffered by the source or target decision function. According to [38], in the following, we set $\tilde{L} = n$, which holds in the worst case.

Theorem 2: When $\beta = (\sqrt{n}/(\sqrt{n} + \sqrt{2\ln 2}))$, we have

$$M \le 2\left(\min\{\Lambda^s, \Lambda\}\right) \tag{22}$$

where $\Lambda^s = \sqrt{2n\ln 2} + \ln 2 + \sum_{i=1}^n \ell_i^s$ and $\Lambda = \sqrt{2n\ln 2} + \ln 2 + \sum_{i=1}^n \ell_i$.

Proof: According to Theorem 1, we have

$$M \le \frac{2}{1-\beta}\left(\ln 2 - (\ln\beta)\min\{\sum_{i=1}^n \ell_i^s, \sum_{i=1}^n \ell_i\}\right).$$

From Lemma 1, the theorem follows immediately. □

The value of β suggested in Theorem 1 will be used in our experiments.

A. Empirical Study of Theoretical Bounds

We use a sample task based on a text-image data set (see the description in Section V) to empirically study the theoretical bounds of OHTHE-SIG. We set $\theta_1 = \theta_1^s = (1/2)$ and $\beta = (\sqrt{n}/(\sqrt{n} + \sqrt{2\ln 2}))$.

Fig. 5 presents the results that are related to the loss bound in Proposition 1. The result of OHTHE-SIG (i.e., the red line)

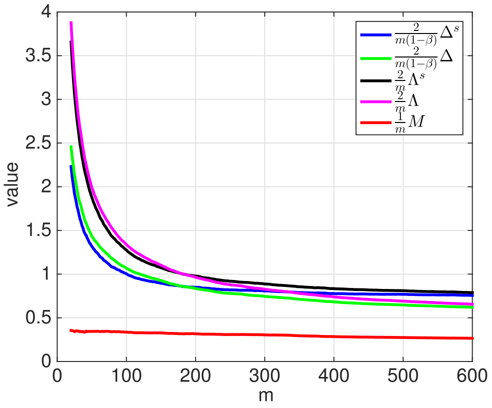


Fig. 6. Results related to the mistake bounds in Theorems 1 and 2, where m is the number of target instances that have been received. The blue and green lines represent the results related to Theorem 1. The black and magenta lines represent the results related to Theorem 2. The red line represents the result produced by OHTHE-SIG.

TABLE I
STATISTICS OF THE DATA SETS IN TERMS OF #INSTANCES × #FEATURES

	text-image data set	cross-language data set	video data set
target data	600×500	$O(10^3) \times O(10^4)$	$20,000 \times 1,024$
source data	$1,200 \times 1,000$	$O(10^3) \times O(10^4)$	$5,000 \times 1,000$
co-occurrence data	$1,600 \times 1,500$	$O(10^3) \times O(10^4)$	$5,000 \times 2,024$

is always the lowest, which conforms to the loss bound in Proposition 1.

Fig. 6 presents the results that are related to the mistake bounds in Theorems 1 and 2. The result of OHTHE-SIG (i.e., the red line) is consistently lower than the others, which verifies the upper bounds in Theorems 1 and 2.

V. EXPERIMENTS

We conduct experiments on three real-world data sets. Table I lists the statistical information of the data sets in terms of #instances × #features, e.g., 600 × 500 in the table means that the text-image data set has 600 target instances, each of which is represented by a 500-D feature vector.

1) *Text-Image Data Set:* This data set is extracted from the NUS-WIDE data set, which is collected from Flickr [53], and includes images and tags posted by users. We refer to the images as the target data and the text instances as the heterogeneous source data. Additionally, images and their corresponding tags are used as the co-occurrence data.

   Since we focus on binary classification tasks, we first select ten classes from the NUS-WIDE data set (each class contains 300 images) and then use images with two classes (e.g., river and sky) to conduct a binary classification task, where the data consist of images with either the positive class (e.g., river) or the negative class (e.g., sky). This setting is also used in the experiments reported in [11]–[13], [15]–[17], and [20]. The number of the generated tasks is $\binom{10}{2} = 45$.
2) *Cross-Language Data Set:* The cross-language data set [54] includes original documents (e.g., the documents written in English) and their translated versions (e.g., the documents that are translated from French to English). We use four languages and six classes to construct $\binom{4}{2} \times \binom{6}{2} = 90$ two-language binary classification tasks, each of which takes two languages as the source and target domains. Specifically, for the English–French binary classification task considering the C15 and CCAT categories, half of the original English documents with C15 or CCAT labels are referred to as the source data, and half of the original French documents with C15 or CCAT labels are referred to as the target data. The remaining original documents and their translated versions are referred to as the co-occurrence data.
3) *Video Data Set:* We also construct a large-scale binary classification task based on the YouTube Multi-view Video Games data set [55], which includes text, audio, and visual features. We take a visual view based on the color histogram as the source feature and a text view based on latent Dirichlet allocation as the target feature. For more details regarding the feature descriptions, please refer to [55]. The objective is to judge whether a target instance is related to a popular video game or not.

A. Baseline Methods

1) *PA:* Online passive-aggressive (PA) algorithm is a traditional online learning algorithm [40]. We adopt PA as a baseline method without knowledge transfer.
2) *SCW:* Soft confidence-weighted (SCW) algorithm is a second-order online learning algorithm [46], which assumes that the linear decision vector is drawn from a Gaussian distribution. SCW is considered the state-of-the-art online learning algorithm without knowledge transfer.
3) *SVM:* SVM is an offline baseline method without knowledge transfer [56]. To handle online target data, we periodically retrain the classifier after receiving $(n/20)$ target instances, where n is the number of target data.
4) *HTL-PA:* HTL-PA finds the nearest neighbors of each target instance in the co-occurrence data and uses the heterogeneous views of the neighbors as the new representation of the target instance. After that, the PA algorithm is performed on the heterogeneous features.
5) *HTLIC-PA:* HTL for image classification (HTLIC) [15] uses heterogeneous source data and co-occurrence data to construct new features for target data. We adjust HTLIC to the online setting by applying PA to the new features.

For fair comparison and simplicity, we set the regularization parameter $c = 1$ for all the algorithms and adopt the linear kernel on all the algorithms except HTLIC-PA. The new representations constructed by HTLIC-PA are dense and low dimensional; thus, they tend to be linearly inseparable. Therefore, we conduct HTLIC-PA using the linear kernel and the Gaussian kernel with $\sigma = \{2^{-5}, 2^{-4}, \ldots, 2^4, 2^5\}$, and report the best result that is achieved when $\sigma = 2^{-5}$. We set the decay factor $\beta = (\sqrt{n}/(\sqrt{n} + \sqrt{2\ln 2}))$ as suggested in Theorem 2, and the number of the nearest neighbors

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

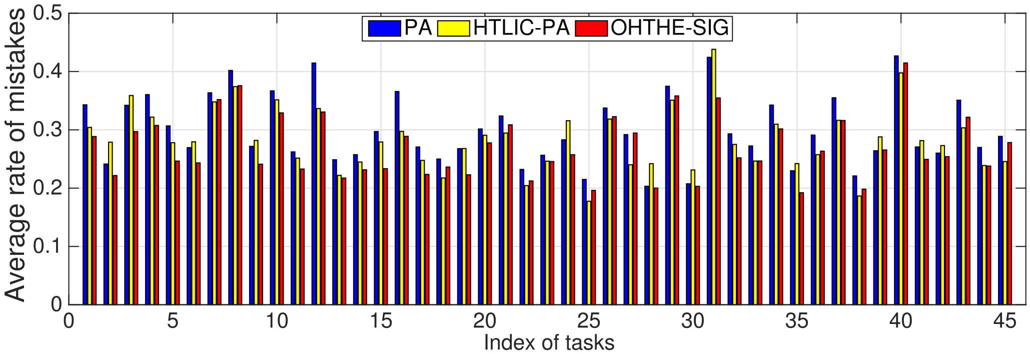
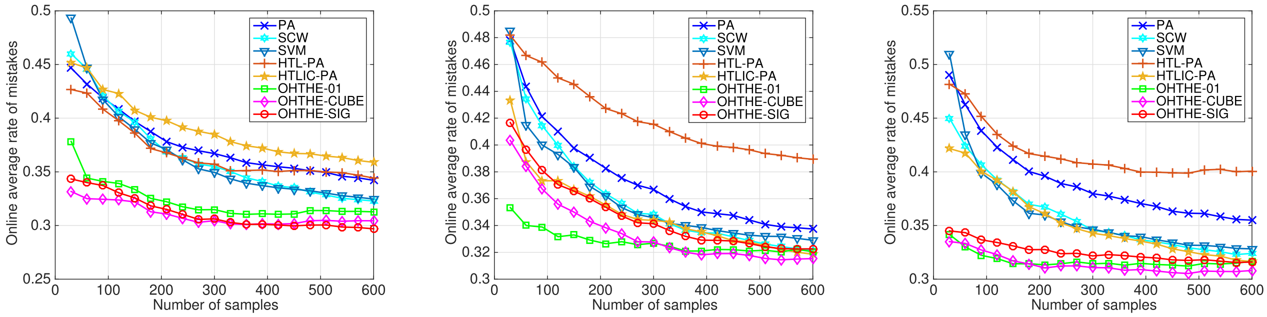


Fig. 7. Average rates of mistakes of different algorithms on the text-image data set, obtained by repeating each task 20 times.

TABLE II

AVERAGE RATES OF MISTAKES ON EXAMPLE TASKS ON THE TEXT-IMAGE DATA SET

Algorithm	task 6	task 15	task 21	task 24	task 33	task 36	Average
PA	0.2695 ± 0.0175	0.2971 ± 0.0143	0.3238 ± 0.0151	0.2829 ± 0.0130	0.2723 ± 0.0158	0.2910 ± 0.0127	0.2997
SCW	0.2449 ± 0.0116	0.2747 ± 0.0114	0.2979 ± 0.0130	0.2543 ± 0.0100	0.2508 ± 0.0115	0.2704 ± 0.0113	0.2788
SVM	0.2532 ± 0.0137	0.2701 ± 0.0122	0.3042 ± 0.0139	**0.2421 ± 0.0129**	0.2491 ± 0.0147	0.2687 ± 0.0134	0.2844
HTL-PA	0.3165 ± 0.0151	0.3076 ± 0.0119	0.3832 ± 0.0143	0.3084 ± 0.0146	0.3055 ± 0.0138	0.3485 ± 0.0076	0.3363
HTLIC-PA	0.2794 ± 0.0100	0.2792 ± 0.0105	0.2945 ± 0.0166	0.3157 ± 0.0120	**0.2465 ± 0.0112**	0.2574 ± 0.0069	0.2834
OHTHE-01	0.2684 ± 0.0090	0.2374 ± 0.0013	**0.2491 ± 0.0014**	0.2907 ± 0.0124	0.2739 ± 0.0089	0.2698 ± 0.0044	0.2834
OHTHE-CUBE	0.2586 ± 0.0065	0.2373 ± 0.0026	0.3015 ± 0.0090	0.2668 ± 0.0113	0.2546 ± 0.0110	**0.2523 ± 0.0100**	0.2759
OHTHE-SIG	**0.2432 ± 0.0114**	**0.2334 ± 0.0091**	0.3086 ± 0.0130	0.2573 ± 0.0107	0.2466 ± 0.0127	0.2634 ± 0.0097	**0.2698**



(a)

(b)

(c)

Fig. 8. Online average rate of mistakes on example tasks on the text-image data set. (a) Task 3. (b) Task 26. (c) Task 37.

$k = (n_c/10)$, where n_c is the number of co-occurrence data. The sensitivities of the parameters c, k, and β will be examined later. In order to obtain stable results, we draw 20 random permutations of the data set and evaluate the performance of each algorithm based on the average rate of mistakes. All the experiments are conducted on a machine with Intel Xeon 3-GHz CPU and 128 GB of RAM, and the source code is implemented in MATLAB.

B. Results of the Text-Image Data Set

Due to the restricted space and observability, we exhibit the results of PA, HTLIC-PA, and OHTHE-SIG for each task in Fig. 7 while leaving out the results of the other algorithms. The x-axis of the figure refers to the 45 tasks. On most tasks, HTLIC-PA and OHTHE-SIG achieve better performances than PA, which indicates that knowledge transferred from the source domain is helpful for the target task. On some tasks, OHTHE-SIG outperforms HTLIC-PA, and on other tasks, OHTHE-SIG is highly comparable with HTLIC-PA. This observation demonstrates the effectiveness of our approach of heterogeneous knowledge transfer.

Table II presents the numerical results of all the used algorithms on several randomly selected tasks and the average results over all the 45 tasks. On average, OHTHE-SIG gets the best result. HTLIC-PA utilizes auxiliary information, such as source and co-occurrence data, to construct new representations that are more effective for classifying the target data. Therefore, HTLIC-PA achieves competitive performance. SVM retrains the classifier and conducts extensive batch training on the target data that have been received. The performance of SVM benefits from more training data. From Fig. 8, we observe that the mistake curves of SVM decrease quickly when more target data have been received. Therefore, when sufficient training data are provided, SVM achieves comparable performance with HTLIC-PA and OHTHE-01. SCW outperforms the first-order method PA, which validates the effectiveness of the second-order method.

Fig. 8 shows the detailed learning processes on three representative tasks. The OHTHE algorithms consistently achieve better results than or highly comparable results with the baseline methods. Moreover, the OHTHE algorithms usually obtain better results at the beginning, which demonstrates the

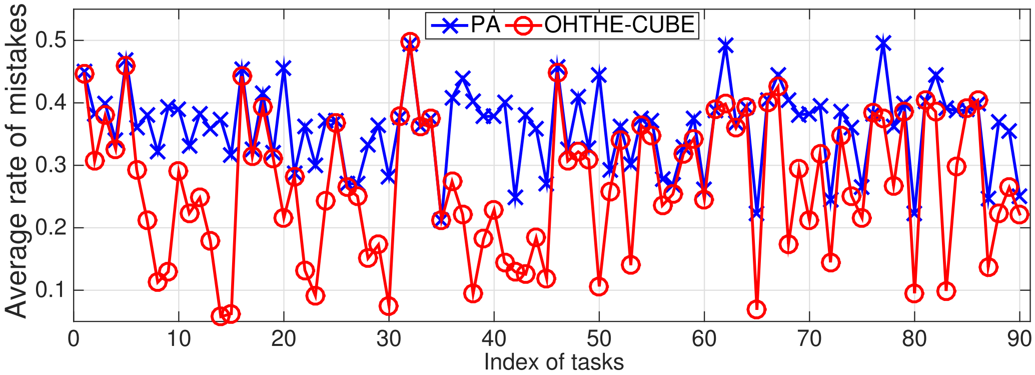


Fig. 9. Average rates of mistakes of different algorithms on the cross-language data set, obtained by repeating each task 20 times.

TABLE III

AVERAGE RATES OF MISTAKES ON EXAMPLE TASKS ON THE CROSS-LANGUAGE DATA SET

Algorithm	task 2	task 33	task 42	task 56	task 66	task 78	Average
PA	0.3821 ± 0.0065	**0.3615 ± 0.0084**	0.2482 ± 0.0057	0.2777 ± 0.0068	0.4042 ± 0.0135	0.3629 ± 0.0117	0.3614
SVM	**0.2176 ± 0.0401**	0.3654 ± 0.0084	0.2208 ± 0.0231	0.2828 ± 0.0109	**0.3975 ± 0.0118**	0.3677 ± 0.0128	0.3226
HTL-PA	0.4243 ± 0.0062	0.3827 ± 0.0090	0.2498 ± 0.0056	0.4794 ± 0.0062	0.4118 ± 0.0135	0.3846 ± 0.0114	0.4345
OHTHE-01	0.3068 ± 0.0003	0.3629 ± 0.0081	**0.1271 ± 0.0004**	0.2642 ± 0.0003	0.4088 ± 0.0162	0.2726 ± 0.0010	0.2699
OHTHE-CUBE	0.3065 ± 0.0002	0.3621 ± 0.0082	0.1295 ± 0.0010	**0.2356 ± 0.0045**	0.4010 ± 0.0129	**0.2672 ± 0.0060**	**0.2670**
OHTHE-SIG	0.3034 ± 0.0015	0.3638 ± 0.0085	0.1553 ± 0.0042	0.2481 ± 0.0060	0.4017 ± 0.0132	0.3132 ± 0.0156	0.2821

TABLE IV

AVERAGE RATES OF MISTAKES AND RUNNING TIMES ON THE VIDEO DATA SET

algorithm	average rate of mistakes	running time (s)
PA	0.4425 ± 0.0028	**0.4073 ± 0.0164**
SCW	0.4434 ± 0.0032	30.5017 ± 0.6009
SVM	0.4430 ± 0.0029	1158.8455 ± 11.8086
OHTHE-01	0.4442 ± 0.0030	66.5473 ± 1.4916
OHTHE-CUBE	0.4408 ± 0.0030	70.2392 ± 2.8647
OHTHE-SIG	**0.4398 ± 0.0030**	68.1423 ± 3.4167

effectiveness of our approach of heterogeneous knowledge transfer. Similar observations are drawn from the other learning tasks.

C. Results of the Cross-Language Data Set

The feature dimension of the cross-language data set is much higher than that of the above text-image data set; thus, SCW and HTLIC-PA require relatively excessive running times to finish the learning tasks under the available resources, and we do not compare them on this data set.

Fig. 9 shows the average rate of mistakes of each task, where OHTHE-CUBE outperforms PA on most tasks. Table III shows the numerical results of several representative tasks. The OHTHE algorithms still achieve the best performance. These results validate that our approach of knowledge transfer effectively leverages the source data to assist in the target task.

D. Results of the Video Data Set

To evaluate the scalability of the proposed algorithms, we compare the OHTHE algorithms with the online learning algorithms PA, SCW, and SVM on the large-scale video data set, which includes tens of thousands of target data, and thousands of source and co-occurrence data.

Table IV lists the average rates of mistakes and running times. The online learning algorithms work in rounds. At each round, an online learning algorithm gets a newly arrived instance, outputs a prediction for this instance, and updates the classifier parameters. As suggested in [50] and [51], the time cost over all rounds of label prediction and parameter updating is used as the running time of an online learning algorithm. From Table IV, PA takes the least running time, while SCW and the OHTHE algorithms have comparable running times. The proposed OHTHE algorithms take relatively longer times because of the additional cost to compute the similarity and find the nearest neighbors. SVM takes much longer running time due to the retraining paradigm.

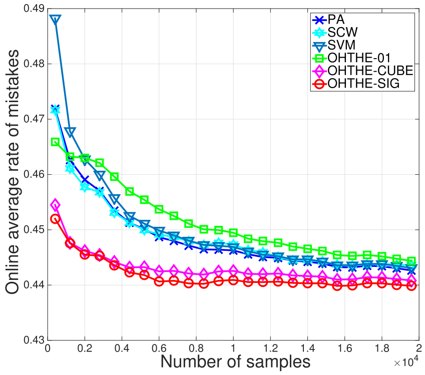
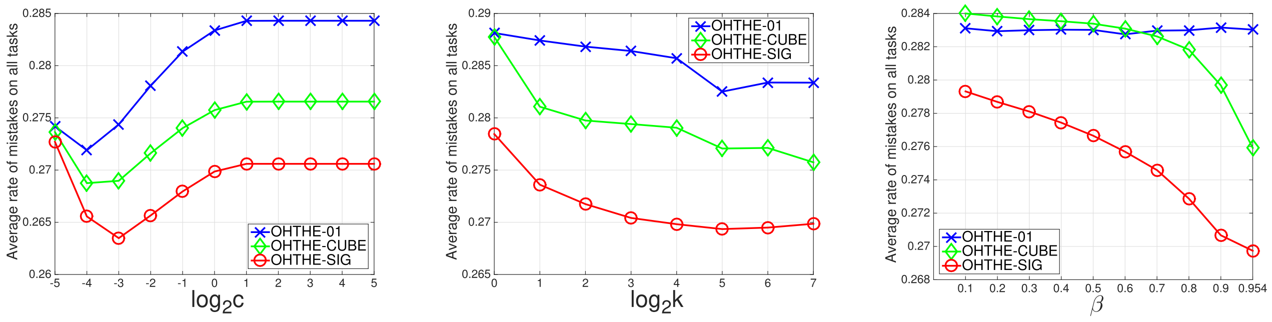


Fig. 10. Online average rates of mistakes on the video data set.

Fig. 10 presents mistake curves with respect to rounds. From Fig. 10, overall PA, SCW, and SVM achieve comparable results, while OHTHE-CUBE and OHTHE-SIG consistently achieve the best results.

E. Parameter Sensitivity

We investigate the influences of the parameters on the performance. The experiments are conducted on the text-image data set, where $n = 600$.



(a) (b) (c)

Fig. 11. Average results over all 45 tasks with different values of the parameters c, k, and β, respectively. (a) Regularization parameter c. (b) Number of nearest neighbors k. (c) Decay factor β.

1) *Effect of the Regularization Parameter c:* Fig. 11(a) shows the average rates of mistakes over all 45 tasks with different values of c, where $k = 2^7 = 128$ and $\beta = (\sqrt{n}/(\sqrt{n} + \sqrt{2\ln 2})) \approx 0.954$. All the OHTHE algorithms have the similar change trend curves. In particular, they achieve their best performances when $c = 2^{-4}$ or 2^{-3}.

2) *Effect of the Number of the Nearest Neighbors k:* Fig. 11(b) shows the average rates of mistakes over all 45 tasks with different values of k, with $c = 2^0 = 1$ and $\beta = (\sqrt{n}/(\sqrt{n} + \sqrt{2\ln 2})) \approx 0.954$. From the figure, the OHTHE algorithms are relatively stable under changes in k when k is greater than $2^3 = 8$.

3) *Effect of the Decay Factor β:* Fig. 11(c) presents the average rates of mistakes over all 45 tasks with different values of β, where $k = 2^7 = 128$ and $c = 2^0 = 1$. From the figure, OHTHE-CUBE and OHTHE-SIG achieve the best results when β is set to the recommended value $(\sqrt{n}/(\sqrt{n} + \sqrt{2\ln 2})) \approx 0.954$.

VI. Conclusion

In this paper, we consider the OHT learning problem, where the target data arrive in an online manner, while the source data and co-occurrence data are from offline sources. In order to connect the source and target domains, we build an offline decision function based on a heterogeneous similarity, which is constructed using the labeled source data and the unlabeled co-occurrence data. After that, an online decision function is built on the target data sequence. Last, we apply a hedge weighting strategy to combine these two kinds of decision functions to boost the learning performance on the target data. We analyze the theoretical bounds of the proposed technique, and perform comprehensive experiments on three real-world data sets to evaluate the proposed algorithms.

In this paper, we address the binary classification problem in the target domain. The multiclass classification problem is more challenging, since it involves learning offline and online decision functions considering multiple classes, and requires a sophisticated strategy to produce an effective combined multiclass classifier. In the future, we plan to focus on exploiting heterogeneous source data to assist in the multiclass classification task in the target domain.

References

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[2] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 53–69, May 2015.

[3] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 1019–1034, May 2014.

[4] P. Wu and T. G. Dietterich, "Improving SVM accuracy by training on auxiliary data sources," in *Proc. ICML*, 2004, p. 110.

[5] X. Liao, Y. Xue, and L. Carin, "Logistic regression with an auxiliary data source," in *Proc. ICML*, 2005, pp. 505–512.

[6] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. ICML*, 2007, pp. 193–200.

[7] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Transferring naive Bayes classifiers for text classification," in *Proc. AAAI*, 2007, pp. 540–545.

[8] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[9] L. Cheng and S. J. Pan, "Semi-supervised domain adaptation on manifolds," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2240–2249, Dec. 2014.

[10] S. Li, S. Song, and G. Huang, "Prediction reweighting for domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 7, pp. 1682–1695, Jul. 2017.

[11] L. Yang, L. Jing, J. Yu, and M. K. Ng, "Learning transferred weights from co-occurrence data for heterogeneous transfer learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2187–2200, Nov. 2016.

[12] L. Yang, L. Jing, and M. K. Ng, "Robust and non-negative collective matrix factorization for text-to-image transfer learning," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4701–4714, Dec. 2015.

[13] B. Tan, Y. Song, E. Zhong, and Q. Yang, "Transitive transfer learning," in *Proc. KDD*, 2015, pp. 1155–1164.

[14] L. Niu, X. Xu, L. Chen, L. Duan, and D. Xu, "Action and event recognition in videos by learning from heterogeneous Web sources," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 6, pp. 1290–1304, Jun. 2017.

[15] Y. Zhu *et al.*, "Heterogeneous transfer learning for image classification," in *Proc. AAAI*, 2011, pp. 1304–1349.

[16] M. K. Ng, Q. Wu, and Y. Ye, "Co-transfer learning via joint transition probability graph based method," in *Proc. 1st Int. Workshop Cross Domain Knowl. Discovery Web Social Netw. Mining (KDD)*, 2012, pp. 1–2.

[17] Q. Wu, M. K. Ng, and Y. Ye, "Cotransfer learning using coupled Markov chains with restart," *IEEE Intell. Syst.*, vol. 29, no. 4, pp. 26–33, Jul./Aug. 2014.

[18] W. Dai, Y. Chen, G.-R. Xue, Q. Yang, and Y. Yu, "Translated learning: Transfer learning across different feature spaces," in *Proc. NIPS*, 2008, pp. 353–360.

[19] G. Wang, D. Hoiem, and D. Forsyth, "Building text features for object image classification," in *Proc. CVPR*, Jun. 2009, pp. 1367–1374.

[20] B. Tan, E. Zhong, M. K. Ng, and Q. Yang, "Mixed-transfer: Transfer learning over mixed graphs," in *Proc. SDM*, 2014, pp. 208–216.

[21] E. Eaton and M. desJardins, "Selective transfer between learning tasks using task-based boosting," in *Proc. AAAI*, 2011, pp. 338–342.

[22] A. Arnold, R. Nallapati, and W. W. Cohen, "A comparative study of methods for transductive transfer learning," in *Proc. 7th IEEE Int. Conf. Data Mining Workshops (ICDM)*, Oct. 2007, pp. 77–82.

[23] Z. Wang, Y. Song, and C. Zhang, "Transferred dimensionality reduction," in *Proc. ECML/PKDD*, 2008, pp. 550–565.

[24] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Self-taught clustering," in *Proc. ICML*, 2008, pp. 200–207.

[25] Z. Ding, M. Shao, and Y. Fu, "Deep low-rank coding for transfer learning," in *Proc. IJCAI*, 2015, pp. 3453–3459.

[26] M. Shao, D. Kit, and Y. Fu, "Generalized transfer subspace learning through low-rank constraint," *Int. J. Comput. Vis.*, vol. 109, nos. 1–2, pp. 74–93, 2014.

[27] X. Shi, Q. Liu, W. Fan, S. Y. Philip, and R. Zhu, "Transfer learning on heterogenous feature spaces via spectral transformation," in *Proc. ICDM*, 2010, pp. 1049–1054.

[28] B. Wei and C. Pal, "Heterogeneous transfer learning with RBMs," in *Proc. AAAI*, 2011, pp. 531–536.

[29] W. Pan and Q. Yang, "Transfer learning in heterogeneous collaborative filtering domains," *Artif. Intell.*, vol. 197, pp. 39–55, Apr. 2013.

[30] J. T. Zhou, S. J. Pan, I. W. Tsang, and Y. Yan, "Hybrid heterogeneous transfer learning through deep learning," in *Proc. AAAI*, 2014, pp. 2213–2220.

[31] J. T. Zhou, I. W. Tsang, S. J. Pan, and M. Tan, "Heterogeneous domain adaptation for multiple classes," in *Proc. AISTATS*, 2014, pp. 1095–1103.

[32] T. Nguyen, T. Silander, and T. Y. Leong, "Transferring expectations in model-based reinforcement learning," in *Proc. NIPS*, 2012, pp. 2555–2563.

[33] Q. Yang, Y. Chen, G.-R. Xue, W. Dai, and Y. Yu, "Heterogeneous transfer learning for image clustering via the social Web," in *Proc. ACL/AFNLP*, 2009, pp. 1–2.

[34] G.-J. Qi, C. Aggarwal, and T. Huang, "Towards semantic knowledge propagation from text corpus to Web images," in *Proc. WWW*, 2011, pp. 297–306.

[35] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[36] S. C. Hoi, J. Wang, and P. Zhao, "LIBOL: A library for online learning algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 495–499, 2014.

[37] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, p. 386, 1958.

[38] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[39] S. Shalev-Shwartz, K. Crammer, O. Dekel, and Y. Singer, "Online passive-aggressive algorithms," in *Proc. NIPS*, 2004, pp. 1229–1236.

[40] 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, Dec. 2006.

[41] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. ICML*, 2003, pp. 928–935.

[42] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for SVM," in *Proc. ICML*, 2007, pp. 807–814.

[43] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: Primal estimated sub-gradient solver for SVM," *Math. Program.*, vol. 127, no. 1, pp. 3–30, Mar. 2011.

[44] M. Dredze, K. Crammer, and F. Pereira, "Confidence-weighted linear classification," in *Proc. ICML*, 2008, pp. 264–271.

[45] K. Crammer, M. Dredze, and F. Pereira, "Exact convex confidence-weighted learning," in *Proc. NIPS*, 2009, pp. 345–352.

[46] S. C. H. Hoi, J. Wang, and P. Zhao, "Exact soft confidence-weighted learning," in *Proc. ICML*, 2012, pp. 1–8.

[47] A. Lazaric and E. Brunskill, "Sequential transfer in multi-armed bandit with finite set of models," in *Proc. NIPS*, 2013, pp. 2220–2228.

[48] L. Ge, J. Gao, and A. Zhang, "OMS-TL: A framework of online multiple source transfer learning," in *Proc. CIKM*, 2013, pp. 2423–2428.

[49] B. Wang and J. Pineau, "Online boosting algorithms for anytime transfer and multitask learning," in *Proc. AAAI*, 2015, pp. 3038–3044.

[50] P. Zhao and S. C. Hoi, "OTL: A framework of online transfer learning," in *Proc. ICML*, 2010, pp. 1231–1238.

[51] P. Zhao, S. C. H. Hoi, J. Wang, and B. Li, "Online transfer learning," *Artif. Intell.*, vol. 216, pp. 76–102, Nov. 2014.

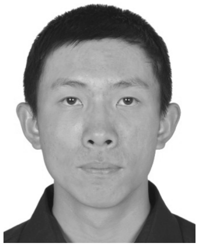
[52] V. Sindhwani and D. S. Rosenberg, "An RKHS for multi-view learning and manifold co-regularization," in *Proc. ICML*, 2008, pp. 976–983.

[53] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng, "NUS-WIDE: A real-world Web image database from national University of Singapore," in *Proc. CIVR*, 2009, p. 48.

[54] M. Amini, N. Usunier, and C. Goutte, "Learning from multiple partially observed views—An application to multilingual text categorization," in *Proc. NIPS*, 2009, pp. 28–36.

[55] O. Madani, M. Georg, and D. A. Ross, "On using nearly-independent feature families for high precision and confidence," *Mach. Learn.*, vol. 92, nos. 2–3, pp. 457–477, 2013.

[56] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear SVM," in *Proc. ICML*, 2008, pp. 408–415.



Yuguang Yan is currently pursuing the Ph.D. degree with the School of Software Engineering, South China University of Technology, Guangzhou, China.

His current research interests include transfer learning, multilabel classification, and online learning.

Qingyao Wu received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2013.

He was a Post-Doctoral Research Fellow with the School of Computer Engineering, Nanyang Technological University, Singapore, from 2014 to 2015. He is currently an Associate Professor with the School of Software Engineering, South China University of Technology, Guangzhou, China. His current research interests include machine learning, data mining, big data research, and bioinformatics.

Mingkui Tan received the Ph.D. degree in computer science from Nanyang Technological University, Singapore, in 2014.

He is currently a Professor with the School of Software Engineering, South China University of Technology, Guangzhou, China. His current research interests include compressive sensing, big data learning, and large-scale optimization.

Michael K. Ng received the B.Sc. and M.Phil. degrees from The University of Hong Kong, Hong Kong, in 1990 and 1992, respectively, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 1995.

He is currently a Chair Professor with the Department of Mathematics, Hong Kong Baptist University, Hong Kong. His current research interests include bioinformatics, image processing, scientific computing, and data mining.

Dr. Ng is a fellow of the Society for Industrial and Applied Mathematics. He serves on the Editorial Board of international journals.

Huaqing Min is currently a Professor with the School of Software Engineering, South China University of Technology, Guangzhou, China. His current research interests include artificial intelligence, machine learning, database, data mining, and robotics.

Ivor W. Tsang received the Ph.D. degree in computer science from The Hong Kong University of Science and Technology, Hong Kong, in 2007.

He is currently an ARC Future Fellow and a Professor with the University of Technology Sydney (UTS), Ultimo, NSW, Australia. He is also the Research Director of the UTS Priority Research Centre for Artificial Intelligence.

Dr. Tsang received the 2008 Natural Science Award (Class II) by the Ministry of Education, China, in 2009, and the prestigious IEEE Transactions on Neural Networks Outstanding 2004 Paper Award in 2007 and the 2014 IEEE Transactions on Multimedia Prize Paper Award.