Online Transfer Learning with Multiple Homogeneous or Heterogeneous Sources

Qingyao Wu, Hanrui Wu, Xiaoming Zhou, Mingkui Tan, Yonghui Xu, Yuguang Yan, and Tianyong Hao

Abstract—Transfer learning techniques have been broadly applied in applications where labeled data in a target domain are difficult to obtain while a lot of labeled data are available in related source domains. In practice, there can be multiple source domains that are related to the target domain, and how to combine them is still an open problem. In this paper, we seek to leverage labeled data from multiple source domains to enhance classification performance in a target domain where the target data are received in an online fashion. This problem is known as the online transfer learning problem. To achieve this, we propose novel online transfer learning paradigms in which the source and target domains are leveraged adaptively. We consider two different problem settings: homogeneous transfer learning and heterogeneous transfer learning. The proposed methods work in an online manner, where the weights of the source domains are adjusted dynamically. We provide the mistake bounds of the proposed methods and perform comprehensive experiments on real-world data sets to demonstrate the effectiveness of the proposed algorithms.

Index Terms—Online transfer learning, multiple source domains, heterogeneous transfer

1 Introduction

Transfer learning is an important research topic in data mining and machine learning and has been extensively studied for many years [1]. The main objective of transfer learning is to make use of labeled data from one or multiple source domains to enhance the learning performance on a target domain in which labeled data for training are difficult to collect. Leveraging knowledge from the labeled source data can dramatically reduce expensive data-labeling efforts.

Most existing studies of transfer learning work on offline batch learning settings [2], [3], in which all the training instances of target domain are assumed to be given in advance. This assumption, however, may not hold in many real-world applications where collecting sufficient training data at one time could be very expensive. Moreover, in some situations, training instances could be received in a sequential manner. One of the simplest approaches to handling online data sequence is to periodically conduct a batch learning algorithm when new data are received. However, this learning paradigm may have a fairly intensive training cost and thus is difficult to be applied in certain actual applications involving large-scale data. As a result, an online learning algorithm that can respond immediately is needed.

Recently, online transfer learning [4], [5], [6], [7], [8] has attracted a lot of attention in the community of machine learning. In the online transfer learning problem, we aim at performing an online learning task in a target domain by leveraging knowledge from some offline source data. An online learning algorithm [9], [10] sequentially updates a classifier based on the feedbacks of a data sequence by processing each instance upon its arrival. The classifier first receives an instance at each round, makes a prediction and obtains the ground-truth label. Then the classifier is updated based on the loss information according to the predicted result and the true label.

To enhance the performance of the target task, we leverage knowledge extracted from offline source data. Instead of focusing solely on a single source domain, we investigate online transfer learning with multiple source domains. We consider two different situations: (i) homogeneous transfer learning where the source and target data are in the same feature space; (ii) heterogeneous transfer learning where the source and target data are represented in different feature spaces.

Take an online news categorization application for example. As shown in Fig. 1, to classify news of NBC, which is regarded as a target domain, we can collect training data from other media websites including BBC, CCTV, and CBS, each of which is regarded as a source domain. Since all the news from these four websites is written in English, this is a homogeneous transfer learning task with three source domains. However, the news from different websites may be different in word preference and expression style. For instance, the BBC news uses typical British English, while NBC uses American English. Therefore, the data distributions of different websites differ from each other, and directly training a classifier on all the source data without knowledge transfer may not achieve satisfying performance. To address this, a more refined transfer learning algorithm with multiple source domains is needed.

- *Q. Wu, H. Wu, X. Zhou, M. Tan, Y. Xu, and Y. Yan are with the School of Software Engineering, South China University of Technology, Guangzhou, Shi 510640, China. E-mail: {qyw, mingkuitan}@scut.edu.cn, {hrwu, ygyan}@outlook.com, miniming.200@gmail.com, xu.yonghui@hotmail.com.*
- *T. Hao is with the School of Informatics, Guangdong University of Foreign Studies, Guangzhou 510420, China. E-mail: haoty@gdufs.edu.cn.*

Manuscript received 24 Feb. 2016; revised 5 Mar. 2017; accepted 15 Mar. 2017. Date of publication 21 Mar. 2017; date of current version 1 June 2017.
Recommended for acceptance by X. Li.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2017.2685597

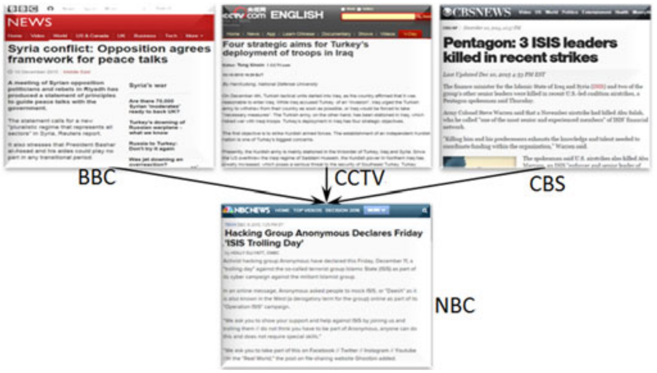

Syria conflict: Opposition agrees framework for peace talks
Four strategic aims for Turkey's deployment of troops in Iraq
Pentagon: 3 ISIS leaders killed in recent strikes
BBC
CCTV
CBS
Hacking Group Anonymous Declares Friday 'ISIS Trolling Day'
NBC

Fig. 1. A multi-source homogeneous transfer learning for news categorization from CCTV, BBC, CBS to NBC.

For the heterogeneous setting [11], the source domain data and target domain data are in different feature spaces. A typical example is shown in Fig. 2. Assuming the feature space of the target domain data is split into two sections, the source domain and the target domain share homogeneous features in the first section while the other section contains heterogeneous features. That is, the feature space of the source domain is a subset of the feature space of the target domain. Such settings hold for a variety of real-world applications. An example of heterogeneous multi-source transfer learning task is image classification in computer vision area where the acquisition of the labeled images of target domain is expensive. However, we can collect labeled images containing a subset of features of the target image from several source domains as the heterogeneous source data.

As shown in Fig. 2, the container (the image of 1st source domain), the truck head (the image of 2nd source domain) and the wheel (the image of 3rd source domain) are parts of a truck (the target domain image). We design a heterogeneous online transfer learning algorithm with multiple sources (HetOTLMS) in this setting. In this algorithm, every new-coming instance of target domain is split into two instances. The first one shares the same feature space with the instance of the source domain; while the other is considered to train a new learner on the target domain. The similar strategy is adopted to adjust weights of source domain and target domain. Finally, the last classifier is also associated with weighted source learners and target learners.

The major contributions of this paper are listed as follows:

- we propose novel online transfer learning algorithms with multiple source domains for both homogeneous and heterogeneous transfer learning tasks;
- we analyze the theoretical properties of the proposed algorithms;
- we validate the effectiveness of the proposed methods by conducting extensive experiments on real-world data sets.

The rest of the paper is organized as follows. In Section 2, we review related works. In Section 3, we provide the problem definition of online transfer learning with multiple sources. In Sections 4 and 5, we present the proposed algorithms for homogeneous and heterogeneous online transfer learning tasks, respectively. Section 6 discusses the experimental results and Section 7 concludes the work.

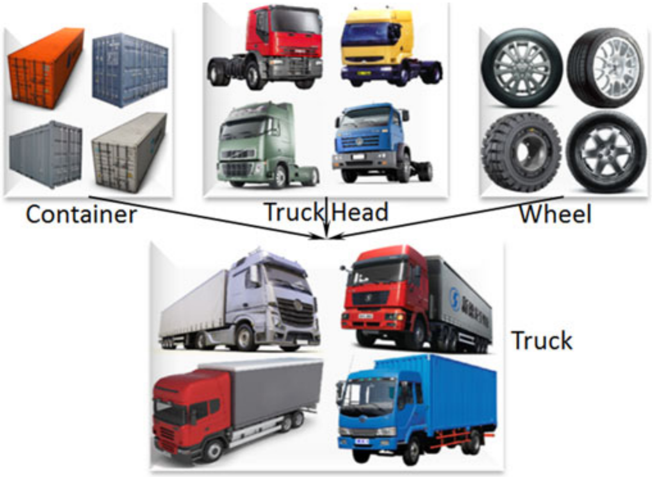


Fig. 2. A heterogeneous multi-source transfer learning task for image categorization from truck head, container, and wheel to truck.

2 Related Work

Online learning has been extensively investigated in the last decade. One of the most famous approaches is the Perceptron algorithm [12], which updates the classifier when an incoming sample is misclassified. A number of works study online learning based on the criterion of maximum-margin [13], [14], [15], [16]. The Double Updating Online Learning (DUOL) algorithm [17] updates both the weight of the current instance and the weights of the instances that have been received. Recently, confidence weighted online learning algorithms have been proposed to utilize the second-order information to update the classifier [18], [19].

Transfer learning aims to make use of knowledge extracted from one or multiple source domains to enhance the performance of a learning task in a target domain [1], [20]. The effectiveness of transfer learning has been demonstrated in many real-world applications, e.g., text mining [21], computer vision [22], recommendation systems [11], etc. [23] proposes a framework for transfer learning based on the structural risk minimization principle and the regularization theory. Different from [23], Domain Transfer Multiple Kernel Learning (DTMKL) [24] attempts to utilize the kernel method into minimizing the structural risk and the distribution mismatch between the source and target domains. Transitive Transfer Learning (TTL) [25] transfers knowledge via intermediate domains with different transferring weights.

To leverage knowledge from multiple source domains, researchers develop boosting-based algorithms to adjust weights of different domains or instances [26], [27], [28], [29]. Among them, [26] performs boosting at the instance level and the domain level to adjust the weights. Yao and Doretto [27] introduces a two-phase training approach, which first summarizes the knowledge from multiple source domains, and then transfers knowledge into the target domain. By this way, [27] achieves better performance on some benchmark data sets. However, [27] considers the homogeneous situation only. Recently, [28] leverages different views from different source domains to assist the target task. Jiang et al. [30] proposes a general framework to preserve the independent information among different tasks. In this paper, we also investigate transfer learning with multiple source domains. Nevertheless, these existing studies deal with batch learning

TABLE 1
Summary of Frequently Used Mathematical Notations

Notations	Mathematical Meanings
D^{S_i}	the ith source domain
n	the number of the source domains
$\mathcal{X}^{S_i}$	the feature space of the ith source domain $\mathcal{X}^{S_i} = \mathbb{R}^{d_i}$
D^S	the set of the source domains $D^S = \{D^{S_i}\}_{i=1}^n$
D^T	target domain
$\mathcal{X}$	the feature space of the target domain $\mathcal{X} = \mathbb{R}^d$
$\mathcal{Y}$	the label space $\mathcal{Y} = \{-1, +1\}$
(x_t, y_t)	the tth labeled instance in the target domain
$f^{S_i}(\cdot)$	the classifier of the ith source domain
$f^T(\cdot)$	the classifier of the target domain
C	the tradeoff parameter
β	the weight discount parameter
$I(\cdot)$	the indicator function

problems where target data are available in advance, while we address transfer learning problems where target data arrive in an online fashion.

Recently, several works studied transfer learning in the framework of online learning. Zhao et al. [4], Zhao and Hoi [31] transfer knowledge from an offline source domain to assist an online learning problem in a target domain. In their works, some offline source data are collected in advance to enhance the performance on target data arriving in an online fashion. However, they consider one source domain only as the auxiliary information. In some real-world applications, auxiliary data can be easily collected from multiple source domains. Motivated by this, in this paper, we propose to exploit knowledge from multiple source domains. We investigate online transfer learning with multiple source domains in two different settings: homogeneous transfer learning and heterogeneous transfer learning. Online transfer learning with homogeneous source domains is devoted to performing a target task by leveraging knowledge from multiple source domains with the same feature space. On the contrary, in online transfer learning with heterogeneous source domains, we consider a situation where the feature spaces of the source domains are different from that of the target domain. In such a situation, to transfer knowledge from multi-source domains to a target domain, we need to find a method of handling the heterogeneous feature space between each source domain and target domain. From these aspects, the problems we are dealing with are more challenging than traditional online transfer learning and multi-source transfer learning.

3 Problem Formulation and General Learning Paradigm

In this section, we formulate the problem of online transfer learning with multiple sources. The learning problem is formally defined as follows, with the notations shown in Table 1. Given n source domains denoted by $D^S = \{D^{S_1}, D^{S_2}, \ldots, D^{S_n}\}$, and a sequence of labeled instances $\{(x_t, y_t) | t = 1, 2, \ldots m\}$ from the target domain D^T. For the ith source domain D^{S_i}, $\mathcal{X}^{S_i} \times \mathcal{Y}$ denotes the source data space, where the feature space $\mathcal{X}^{S_i} = \mathbb{R}^{d_i}$, and the label space $\mathcal{Y} = \{-1, +1\}$. f^{S_i} represents the classifier learned on the ith source domain.

Let $\mathcal{X} \times \mathcal{Y}$ denote the data space of the target domain, where the feature space $\mathcal{X} = \mathbb{R}^d$ and the label space $\mathcal{Y} = \{-1, +1\}$. Note that the target and source domains share the same label space $\mathcal{Y}$. If sufficient target data are available, a stable classifier regarding the target domain can be learned by solving the following optimization problem

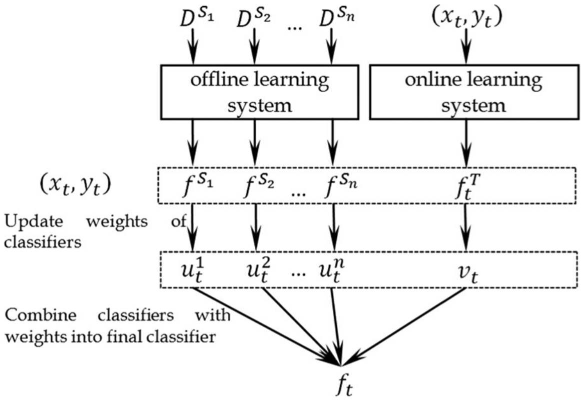

D^{S_1} D^{S_2} ... D^{S_n}
(x_t, y_t)
offline learning system
online learning system
(x_t, y_t)
f^{S_1} f^{S_2} ... f^{S_n}
f_t^T
Update weights of classifiers
u_t^1 u_t^2 ... u_t^n
v_t
Combine classifiers with weights into final classifier
f_t

Fig. 3. The structure of the homogeneous online learning paradigm.

$$\min_{f \in \mathcal{H}} \frac{1}{2} \left\| f^T \right\|^2 + \lambda \sum_{t=1}^{m} \ell\big(f^T(x_t), y_t\big), \tag{1}$$

where $\mathcal{H}$ denotes the reproducing kernel Hilbert space defined by kernel $k(\cdot, \cdot)$, λ is a trade-off parameter and $\ell\big(f^T(x_t), y_t\big) = \max\big(1 - y_t f^T(x_t), 0\big)$ denotes the hinge loss function. Here the kernel function $k(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is defined for the target domain. According to the Representer Theorem [32] and the traits of online learning, the target classifier can be represented by

$$f^T(x) = \sum_{i=1}^{t} \alpha_i y_i k(x_i, x), \tag{2}$$

where α_i is the coefficient of the ith target instance. Relying on the online learning scheme, we can easily learn the target classifier via the online Passive-Aggressive algorithm (PA) [16]. Specifically, if the classifier suffers a non-zero loss, i.e., $1 - y_t f^T(x_t) > 0$, the instance is regarded as a support vector and will be added into the set of support vectors, i.e., $f^T := f^T + \alpha_t y_t k(x_t, \cdot)$, where $\alpha_t = \min\{C, \frac{\ell(f^T(x_t), y_t)}{k(x_t, x_t)}\}$ is the coefficient. The parameter C prevents the coefficient of a support vector from being too large.

If the target domain data are insufficient, the above online learning scheme cannot guarantee an effective classifier, and hence knowledge transfer from a source domain becomes necessary. Therefore, we propose a three-stage paradigm to build a classifier on the target domain by exploiting useful information from multiple source domains, as is shown in Figs. 3 and 4, regarding homogeneous transfer learning and heterogeneous transfer learning, respectively. The main idea of this learning paradigm is to combine the multiple classifier models that are built on the sources and target domain for effective online prediction. Mathematically, the combined decision function for the target task can be written as

$$d^T(x, w) = \sum_{i=1}^{t} \alpha_i y_i k(x_i, x) + \sum_{j=1}^{n} w_j f^{s_j}(x_i), \tag{3}$$

where w_j is weight value for the jth source domain classifier.

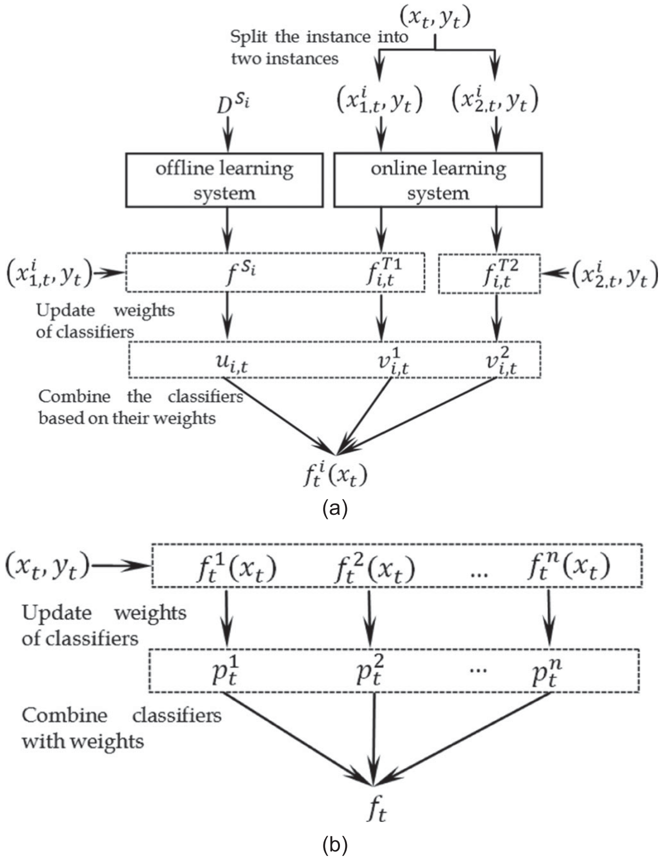


Fig. 4. The structure of the heterogeneous online learning paradigm.

To effectively transfer knowledge from multiple source domains, we propose to combine multiple decision functions, which are learned on different domains, respectively, by a multi-layer strategy. Fig. 3 presents the proposed three-layer strategy for homogeneous transfer learning. The first layer has $n+1$ decision functions, where f^{S_i} is the decision function learned on the ith source domain, and f^T is the target decision function. In the second layer, each decision function is associated with a weight, which reflects the contribution it makes to the final prediction. In the third layer, we combine the source and target decision functions based on their weights to build the final classifier f_t. At each round, we update the target decision f^T and the weights of all the decision functions according to the predicted result and the ground-truth label of the target instance.

For the heterogeneous transfer learning, the source and target domains could have different feature spaces. Therefore, heterogeneous transfer learning is generally more challenging than homogeneous transfer learning. It is difficult, if not impossible, to transfer knowledge in heterogeneous transfer learning when the feature spaces of the source and target domains do not share any feature at all. To simplify the problem, similar to the problem setting in [4] we assume that the feature space of each source domain is a subset of that of the target domain, i.e., $\mathbb{R}^{d_i} \subseteq \mathbb{R}^d$, where $\mathbb{R}^{d_i}$ is the feature space of the ith source domain, and $\mathbb{R}^d$ is the feature space of the target domain.

Furthermore, the feature spaces of the source domains could be different from each other. As shown in Fig. 5, the dimension of the target domain D^T in this toy example is 100, where the first 30 ones share the same feature space with that of the source domain D^{S_1}, while the features from 41 to 90 are mapped to that of the second source domain D^{S_2}, and the last 25 dimensions are identical to that of source domain D^{S_3}. Formally, let $(\mathcal{X}^{S_1}, \ldots, \mathcal{X}^{S_n})$ be the feature spaces of source domains, we have $\mathcal{X}^{S_i} \neq \mathcal{X}^{S_j}$ for $\forall i \neq j$.

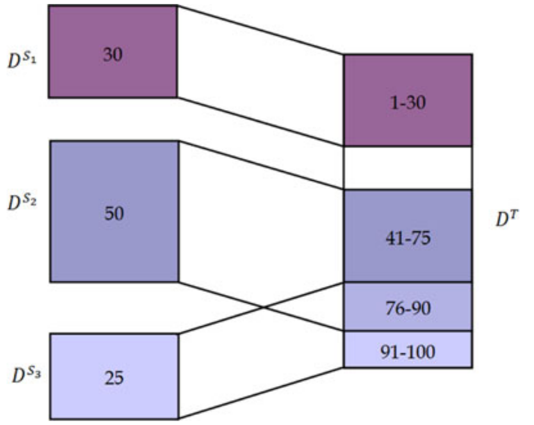


Fig. 5. The relation of feature spaces between source domains and target domain in a toy example.

To address the problem of heterogeneous transfer learning, we design a two-stage learning paradigm, which is shown in Fig. 4. In the first stage (see Fig. 4a), we train a decision function for each source domain. Specifically, for the ith source domain, the feature space of the target domain is split into two parts: a part that is homogeneous with the ith source domain, and the other part that consists of the other features. We build two decision functions $f^{T1}_{i,t}$ and $f^{T2}_{i,t}$ on these two feature subsets. Afterwards, we combine $f^{T1}_{i,t}$ and $f^{T2}_{i,t}$ to obtain a decision function $f^i_t(x_t)$ for the feature subset that corresponds to the ith source domain. In the second stage (see Fig. 4b), we combine the n decision functions that are constructed in the first stage based on their weights to obtain an ensemble classifier.

4 Homogeneous Online Transfer Learning

In this section, we present an algorithm called HomOTLMS, which stands for **Hom**ogeneous **O**nline **T**ransfer **L**earning with **M**ultiple **S**ource domains, where all the domains share the same feature space, i.e., $\mathcal{X}^{S_i} = \mathcal{X}, \forall i$. Motivated by the Hedge method [33], we combine multiple decision functions that are constructed on the source domains, respectively, to build an ensemble classifier for the target instances. Fig. 3 illustrates the learning paradigm of HomOTLMS, and Algorithm 1 presents the pseudo code.

4.1 The Learning Steps of HomOTLMS

As shown in Fig. 3, the source and target decision functions are built separately. Since the source data are given in advance, the decision functions f^{S_1} to f^{S_n} from the source domains can be built in an offline batch learning paradigm. For the target data that arrive in an online fashion, the target decision function f^T is learned by the PA online learning algorithm [16], which constructs a set of support vectors that come from the target instances. The set of support vectors is empty initially, i.e., ($f^T_1 = \emptyset$). At the tth round, f^T_t makes a prediction for the given instance x_t, and then a loss is computed based on the hinge loss function $\ell(y, x, f) = \max(0, 1 - yf(x))$. Specifically, in case that the decision function suffers a non-zero loss value on the instance x_t, we add it as a support vector into the set, i.e., $f_{t+1}^T = f_t^T + \tau_t y_t k(x_t, \cdot)$, where $\tau_t = \min\{C, \ell_t / k(x_t, x_t)\}$ is the coefficient of the support vector.

Algorithm 1. HomOTLMS

```
Input: the classifiers from source domains $f^S = (f^{S_1}, f^{S_2}, \ldots, f^{S_n})$, initial tradeoff $C$, and the weight discount $\beta \in (0,1)$.
Initialize: $f_1^T = \emptyset$, $u_1 = 1/(n+1)$, $v_1 = 1/(n+1)$.
for $t = 1, 2, \ldots, m$. do
    receive instance: $x_t \in \mathcal{X}$.
    $p_t^i = u_t^i / (\sum_{i=1}^n u_t^i + v_t)$, $p_t^v = v_t / (\sum_{i=1}^n u_t^i + v_t)$.
    predict: $\hat{y}_t = \mathrm{sign}(\sum_{i=1}^n p_t^i f^{S_i}(x_t) + p_t^v f_t^T(x_t))$.
    receive correct label: $y_t \in \{-1, +1\}$.
    for $i = 1, 2, \ldots, n$. do
        $z_t^i = I(\mathrm{sign}(y_t f^{S_i}(x_t)) < 0)$.
        $u_{t+1}^i = u_t^i \beta^{z_t^i}$.
    end for
    $z_t^v = I(\mathrm{sign}(y_t f_t^T(x_t)) < 0)$.
    $v_{t+1} = v_t \beta^{z_t^v}$.
    Suffer loss: $\ell_t = [1 - y_t f_t^T(x_t)]_+$.
    if $\ell_t > 0$. then
        $f_{t+1}^T = f_t^T + \tau_t y_t k(x_t, \cdot)$ where $\tau_t = \min\{C, \ell_t / k(x_t, x_t)\}$.
    end if
end for
Output: $f_t(x) = \mathrm{sign}(\sum_{i=1}^n p_t^i f^{S_i}(x) + p_t^v f_t^T(x))$.
```

Then, a weight vector $\mathbf{u}_t = (u_t^1, u_t^2, \ldots, u_t^n)^\top$ and a weight variable v are constructed to represent the contributions of the source and target decision functions, respectively. We apply the Hedge Algorithm [33] to dynamically update the weights of both source and target decision functions to obtain an ensemble classifier.

4.2 The Pseudo-Code of HomOTLMS

We provide the pseudo-code of HomOTLMS in Algorithm 1. Specifically, at the tth round of the target learning task, an instance x_t comes and every classifier makes a prediction. If a classifier makes an incorrect prediction, its weight will decrease with a discount, i.e., $u_{t+1}^i = u_t^i \beta$ for the source decision functions and $v_{t+1} = v_t \beta$ for the target decision function, where $\beta \in (0,1)$ is the weight discount parameter. Based on the constructed decision functions and their weights, the ensemble classifier is given by

$$\hat{y}_t = \mathrm{sign}\left(\sum_{i=1}^n p_t^i f^{S_i}(x_t) + p_t^v f_t^T(x_t) \right), \tag{4}$$

where $p_t^i = u_t^i / (\sum_{i=1}^n u_t^i + v_t)$ and $p_t^v = v_t / (\sum_{i=1}^n u_t^i + v_t)$ are the normalized weights.

4.3 Theoretical Analysis of HomOTLMS

We provide the mistake bound of the algorithm HomOTLMS as follows.

Theorem 1. *Let us denote M as the number of mistakes made by the algorithm HomOTLMS. By choosing $\beta = \sqrt{M_{\min}} / (\sqrt{M_{\min}} + \sqrt{\ln(n+1)})$ we have M bounded by*

$$M \leq M_{\min} + \sqrt{2\ln(n+1) \times M_{\min}} + \ln(n+1), \tag{5}$$

where

$$M_{\min} = \min\{M_s^i, M_t\},$$

$$M_s^i = \sum_{t=1}^m z_t^i,$$

$$M_t = \sum_{t=1}^m z_t^v.$$

Proof. From the Theorem 2 in Hedge Algorithm [33], if we set all the initial weights equally to be $1/(n+1)$ for each source classifier, we can obtain the following mistake bound for M,

$$M \leq \frac{M_{\min} \ln(1/\beta) + \ln(n+1)}{1 - \beta}. \tag{6}$$

It can be shown that $\ln(1/\beta) \leq (1 - \beta^2)/2\beta$ for $\beta \in (0,1]$. If we apply this inequality to (6), then we can obtain

$$\frac{M_{\min} \ln(1/\beta) + \ln(n+1)}{1 - \beta} \leq \frac{M_{\min}(1+\beta)}{2\beta} + \frac{\ln(n+1)}{1-\beta}. \tag{7}$$

By substituting $\beta = \sqrt{M_{\min}} / (\sqrt{M_{\min}} + \sqrt{\ln(n+1)})$ into (7), we obtain (5). □

Theorem 1 provides an upper bound of the mistake for our algorithm. The results in Theorem 1 imply that the learner's average per round mistake can never be much larger than that of the best pure strategy. Since the mistake bound depends logarithmically on n only, this bound is reasonable even for a very large number of strategies.

Algorithm 2. HetOTLMS

```
Input: the classifiers from source domains $f^S = (f^{S_1}, f^{S_2}, \ldots, f^{S_n})$, initial tradeoff $C$, and the weight discount $\beta_1$, $\beta_2$, $\in (0,1)$.
Initialize: $[f_{i,1}^{T1} = \emptyset,\ f_{i,1}^{T2} = \emptyset,\ u_{i,1} = v_{i,t}^1 = v_{i,t}^2 = 1/3,\ w_1^i = 1/n$. (for $i = 1, 2, \ldots n$).
for $t$th iteration, $t = 1, 2, \ldots, m$. do
    Receive instance: $x_t \in \mathcal{X}$.
    Split $x_t$ into two instances: $x_{1,t}^i$, $x_{2,t}^i$.
    $p_t^i = w_t^i / \sum_{i=1}^n w_t^i$.
    Predict: $\hat{y}_t = \mathrm{sign}(\sum_{i=1}^n f_t^i(x_t))$.
    Receive correct label: $y_t \in \{-1, +1\}$.
    for $i$th source, $i = 1, 2, \ldots, n$. do
        Compute suffer loss: $\ell_{1,t} = [1 - y_t f_{i,t}^{T1}(x_{1,t}^i)]_+$.
        if $\ell_{1,t} > 0$. then
            $f_{i,t+1}^{T1} = f_{i,t}^{T1} + \tau_{1,t} y_t k_{i,1}(x_{1,t}^i, \cdot)$.
        end if
        Compute suffer loss: $\ell_{2,t} = [1 - y_t f_{i,t}^{T2}(x_{2,t}^i)]_+$.
        if $\ell_{2,t} > 0$. then
            $f_{i,t+1}^{T2} = f_{i,t}^{T2} + \tau_{2,t} y_t k_{i,2}(x_{2,t}^i, \cdot)$.
        end if
        If $\mathrm{sign}(y_t f^{S_i}(x_{1,t}^i)) < 0$, Then $u_{i,t+1} = u_{i,t} \beta_2$.
        If $\mathrm{sign}(y_t f_{i,t}^{T1}(x_{1,t}^i)) < 0$, Then $v_{i,t+1}^1 = v_{i,t}^1 \beta_2$.
        If $\mathrm{sign}(y_t f_{i,t}^{T2}(x_{2,t}^i)) < 0$, Then $v_{i,t+1}^2 = v_{i,t}^2 \beta_2$.
        If $\mathrm{sign}(y_t f_t^i(x_t)) < 0$, Then $w_{t+1}^i = w_t^i \beta_1$.
        Compute $f_{t+1}^i(x_t)$ by equation (8).
    end for
end for
Output: $f_t(x) = \mathrm{sign}(\sum_{i=1}^n p_t^i f_t^i(x))$.
```

5 Heterogeneous Online Transfer Learning

In this section, we present the proposed algorithm HetOTLMS, which stands for **Het**erogeneous **O**nline **T**ransfer **L**earning with **M**ultiple **S**ource domains. Fig. 4 and Algorithm 2 present the learning paradigm and pseudo code of HetOTLMS, respectively.

5.1 The First Stage of HetOTLMS

Remind that the feature space of a source domain is a subset of that of the target domain. Therefore, in order to exploit knowledge from the source data, for the ith source domain, we split the target feature space into two sections, the first one is homogeneous with the ith source domain, and the other one is the remainder of the target feature space. Based on each source domain S_i, we learn three base classifiers $(f^{S_i}, f^{T1}_{i,t}, f^{T2}_{i,t})$. f^{S_i} is learned by using offline learning algorithms in source domain. $f^{T1}_{i,t}$ and $f^{T2}_{i,t}$ are learned by combining the first section and the second section in target domain with the source domain, respectively. Then, classifier's weights $u_{i,t}, v^1_{i,t}, v^2_{i,t}$ related to the $f^{S_i}, f^{T1}_{i,t}, f^{T2}_{i,t}$ are learned. Combining the base classifiers by the learned weights, we can obtain a strong classifier $f^i_t(x_t)$ for each source domain S_i. Suppose there are n source domains, we can obtain $(f^1_t(x_t), f^i_2(x_t), \ldots, f^i_n(x_t))$. The structure of the first stage is shown in Fig. 4a.

5.2 The Second Stage of HetOTLMS

In the second stage, we combine the n classifiers $(f^1_t(x_t), f^2_t(x_t), \ldots, f^n_t(x_t))$ into an ensemble model to make a final prediction. To achieve this, a weight vector $\mathbf{w} = (w^1, w^2, \ldots, w^n)$ is constructed to represent the weights of n classifiers, and we use the Hedge algorithm to learn the weights of the classifiers in the ensemble. As shown in Fig. 4b, on the tth round, a discount is computed on its weight with a weight discount $\beta_1 \in (0,1)$ when the classifier $f^i_t(x_t)$ suffers a loss value. In this paper, we set $\beta_1 = \sqrt{T}/(\sqrt{T} + \sqrt{\ln n})$. Formally, we have $w^i_{t+1} = w^i_t \beta_1$. The structure of the second stage is shown in Fig. 4b.

5.3 The Pseudo-Code of HetOTLMS

We provide the pseudo-code of HetOTLMS in Algorithm 2. Specifically, we first input the learned $(f^1_t(x_t), f^2_t(x_t), \ldots, f^n_t(x_t))$, and initialized classifiers $f^{T1}_{i,t}$ and $f^{T2}_{i,t}$ as empty $(f^{T1}_{i,1} = \emptyset, f^{T2}_{i,1} = \emptyset)$.

In each iteration (i.e., tth iteration), we split the corresponding data instance x_t into two parts: $x_{1,t} \in \mathbb{R}^{d_i}$ and $x_{2,t} \in \mathbb{R}^d/\mathbb{R}^{d_i}$. After the split, we first predict a label for instance x_t as step 4 in Algorithm 2. Then, for each source domain (i.e., S_i), we update classifiers $f^{T1}_{i,t}$ and $f^{T2}_{i,t}$ as the following roles [16].

Role 1 (steps 8 to 11 in Algorithm 2). If classifier $f^{T1}_{i,t}$ suffers a loss ($\ell_{1,t} = [1 - y_t f^{T1}_{i,t}(x_{1,t})]_+$) based on the hinge loss function. The instance $x^i_{1,t}$ would be considered as a support vector to add into the set of support vectors ($f^{T1}_{i,t+1} = f^{T1}_{i,t} + \tau_{1,t} y_t k_{i,1}(x^i_{1,t}, \cdot)$, where $\tau_{1,t} = \min\{C, \ell_{1,t}/k_{i,1}(x^i_{1,t}, x^i_{1,t})\}$ is the coefficient of the support vector, $k_{i,1}(\cdot,\cdot) : \mathbb{R}^{d_i} \times \mathbb{R}^{d_i} \rightarrow$ indicates the kernel functions for $x^i_{1,t}$).

Role 2 (steps 12 to 15 in Algorithm 2). If classifier $f^{T2}_{i,t}$ suffers a loss ($\ell_{2,t} = [1 - y_t f^{T2}_{i,t}(x_{2,t})]_+$) based on the hinge loss function. The instance $x^i_{2,t}$ would be considered as a support vector to add into the set of support vectors($f^{T2}_{i,t+1} = f^{T2}_{i,t} + \tau_{2,t} y_t k_{i,2}(x^i_{2,t}, \cdot)$, where $\tau_{2,t} = \min\{C, \ell_{2,t}/k_{i,2}(x^i_{2,t}, x^i_{2,t})\}$ is the coefficient of the support vector, $k_{i,2}(\cdot,\cdot) : \mathbb{R}^{d-d_i} \times \mathbb{R}^{d-d_i} \rightarrow$ indicates the kernel functions for $x^i_{2,t}$).

Then, the weighting vector $((u_{\mathrm{i,t}}, v^1_{i,t}, v^2_{i,t}))$ which respect the weights of f^{S_i}, $f^{T1}_{i,t}$, and $f^{T2}_{i,t}$ are learnt to construct an ensemble classifier model $f^i_t(x_t)$ (steps 16 to 18 in Algorithm 2). In this paper, we set $\beta_2 = \sqrt{T}/(\sqrt{T} + \sqrt{\ln 3})$. Then, we update the weight w^i_{t+1} for $f^i_t(x_t)$ as step 19. After updating $f^i_{t+1}(x_t)$, we compute $f^i_{t+1}(x_t)$ by the following equation,

$$f^i_{t+1}(x_t) = \frac{u_{i,t} f^{S_i}(x^i_{1,t}) + v^1_{i,t} f^{T1}_{i,t}(x^i_{1,t}) + v^2_{i,t+1} f^{T2}_{i,t}(x^i_{2,t})}{u_{i,t} + v^1_{i,t} + v^2_{i,t}}. \quad (8)$$

By this way, we can obtain $(f^1_t(x_t), f^2_t(x_t), \ldots, f^n_t(x_t))$ and their weights $((w^1_t, w^2_t, \ldots, w^n_t))$ for each source domain. Then we can predict the label for an input test instance x by the following equation,

$$f_t(x) = \operatorname{sign}\left(\sum_{i=1}^{n} p^i_t f^i_t(x)\right), \quad (9)$$

where $p^i_t = w^i_t / \sum_{i=1}^n w^i_t$ is the normalized weight of w^i_t.

5.4 Theoretical Analysis of HetOTLMS

The ensemble prediction of a new instance x_t is given as Equation (9). We set $\hat{y}_t = f_t(x)$ as the predict label of x, and show the mistake bound for algorithm HetOTLMS as follows.

Theorem 2. *Let us denote M as the number of mistakes made by the algorithm HetOTLMS. By choosing*

$$\beta_1 = \sqrt{\min\{M_i\}}\Big/\left(\sqrt{\min\{M_i\}} + \sqrt{\ln n}\right),$$

we have M bounded by

$$M \leq \min\{M_i\} + \sqrt{2 \ln n \times \min\{M_i\}} + \ln n, \quad (10)$$

where $M_i = \sum_{t=1}^m I\big(\operatorname{sign}\big(y_t f^i_t(x_t)\big) < 0\big)$.

Proof. The proof of Theorem 2 is similar to that of Theorem 1. We just need to replace $\min\{M^i_s, M_t\}$ and β with $\min\{M_i\}$ and β_1, respectively, to get (8). As to the bound M_i, by choosing

$$\beta_2 = \frac{\sqrt{\min\left\{M^i_s, M^{i,1}_t, M^{i,2}_t\right\}}}{\left(\sqrt{\min\left\{M^i_s, M^{i,1}_t, M^{i,2}_t\right\}} + \sqrt{\ln 3}\right)},$$

we obtain M_i bounded from above by $M_i \leq \min\{M^i_s, M^{i,1}_t, M^{i,2}_t\} + \sqrt{2 \ln 3 \times \min\{M^i_s, M^{i,1}_t, M^{i,2}_t\}} + \ln 3,$ where

$$M^i_s = \sum_{t=1}^m I\Big(\operatorname{sign}\Big(y_t f^{S_i}\Big(x^i_{1,t}\Big)\Big) < 0\Big),$$

$$M^{i,1}_t = \sum_{t=1}^m I\Big(\operatorname{sign}\Big(y_t f^{T1}_{i,t}\Big(x^i_{1,t}\Big)\Big) < 0\Big),$$

$$M^{i,2}_t = \sum_{t=1}^m I\Big(\operatorname{sign}\Big(y_t f^{T2}_{i,t}\Big(x^i_{2,t}\Big)\Big) < 0\Big). \qquad \square$$

TABLE 2
Detailed Information of 20Newsgroups Data Set Used in Experiments

Tasks	SD size	TD size	Dimensions	Source Domains (SD)	Target Domain (TD)
os_versus_crypt	5,875	1,952	61,188	ibm_versus_electronics, ibm_versus_electronics, mac_versus_med, x_versus_space	os_versus_crypt
ibm_versus_electronics	5,864	1,963	61,188	os_versus_crypt, mac_versus_med, x_versus_space	ibm_versus_electronics
mac_versus_med	5,882	1,945	61,188	os_versus_crypt, ibm_versus_electronics, x_versus_space	mac_versus_med
x_versus_space	5,860	1,967	61,188	os_versus_crypt, ibm_versus_electronics, mac_versus_med	x_versus_space

TABLE 3
Detailed Information of Sentiment Analysis Data Set Used in Experiments

Tasks	SD size	TD size	Dimensions	Source Domains (SD)	Target Domain (TD)
books	6,000	2,000	473,857	DVDs, electronics, kitchen	books
DVDs	6,000	2,000	473,857	books, electronics, kitchen	DVDs
electronics	6,000	2,000	473,857	books, DVDs, kitchen	electronics
kitchen	6,000	2,000	473,857	books, DVDs, electronics	kitchen

Theorem 2 provides the upper bound on the mistake for HetOTLMS algorithm. Similarly, the results in Theorem 1, the upper bound on a mistake of HetOTLMS can never be much larger than that of the best pure strategy. HetOTLMS can also be used for a large number of strategies since its mistake bound depends only logarithmically on n.

6 Experiments

In this section, we compare the performance of the proposed algorithms with online transfer learning baseline methods on three real-world data sets: the 20Newsgroups data set, the sentiment analysis data set and the five languages data set. The experiments are carried out in homogeneous and heterogeneous settings. To obtain reliable results, we repeat each experiment 20 times by changing the order in which the test instance arrives. We record and report the average results of 20 replicate experiments, and the results show that the proposed algorithms can achieve better performances against the compared algorithms.

6.1 Data Sets

6.1.1 20Newsgroups Data Set

The 20Newsgroups data set[1] consists of a collection of approximately 20,000 newsgroup documents across different topics, which have sub-topics, e.g., comp.os.ms-windows.misc (os for short), comp.sys.ibm.pc.hardware (ibm for short), comp.sys.mac.hardware (mac for short), and comp.windows.x (x for short) are the sub-topics of comp while sci.crypt (crypt for short), sci.electronics (electronics for short), sci.med (med for short), and sci.space (space for short) are the sub-topics of sci. In our experiments, the instances with respect to the sub-topics of comp are labeled as positive instances, and those of the sub-topics of sci are labeled as negative instances, leading to four related learning domains (os_versus_crypt, ibm_versus_electronics, mac_versus_med and x_versus_space). Then, we randomly choose one domain as the target domain, the remaining domains are used as source domains. By this way, we can generate four transfer learning tasks. The characteristics of these four tasks are summarized as Table 2.

Note that the data set generated from 20Newsgroups is composed of several topics and subtopics in this paper. For a task, the positive/negative instance in the target domain and the source domains belong to the same category but different sub-categories. The reason for this setting is because we expect a similar but not the same distribution between different sources. Thereby, we can use the generated data for testing the knowledge transfer ability of our algorithm between different domains.

6.1.2 Sentiment Analysis Data Set

The sentiment analysis data set consists of Amazon product reviews for four different product types: books, DVDs, electronics, and kitchen. Each review consists of a human rating score (0-5 stars), a review title, a product name, a reviewer name, location, date, and the review content. Reviews with rating >3 are labeled as positive instances, those with rating <3 are labeled as negative instances, and the rest are discarded because their polarity is ambiguous. The dimension of the instances is 473,856. Four domains (books, DVDs, electronics, and kitchen) are constructed on the sentiment analysis data set. Each task consists of 2,000 instances and the number of positive instances is the same with that of negative instances. Then, we randomly choose one domain as the target domain, the remain domains are used as source domains. By this way, we can generate four transfer learning tasks. The characteristics of these four tasks are summarized as Table 3.

6.1.3 Five Language Data Set

We use the five languages data set [34] to generate learning tasks. The data set contains feature characteristics of documents written in five different languages (English, French, German, Spanish, and Italian) but sharing the same set of categories. Each language contains indexes of the documents written or translated in that language. For example, English contains files: Index EN-EN (original

[1] http://qwone.com/~jason/20Newsgroups/

TABLE 4
Detailed Information of Five Language Data Set Used in Experiments

Tasks	SD size	TD size	Dimensions	Source Domains (SD)	Target Domain (TD)
EN-EN	31,928	6,309	21,532	FR-EN, GR-EN, IT-EN, SP-EN	EN-EN
FR-FR	28,791	9,446	24,894	EN-FR, GR-FR, IT-FR, SP-FR	FR-FR
GR-GR	28,237	10,000	34,280	EN-GR, FR-GR, IT-GR, SP-GR	GR-GR
IT-IT	31,492	6,745	15,507	EN-IT, FR-IT, GR-IT, SP-IT	IT-IT
SP-SP	32,500	5,737	11,548	EN-SP, FR-SP, GR-SP, IT-SP	SP-SP

TABLE 5
Results (Mean±Standard Deviations) of Applying Different Learning Algorithms on the 20Newsgroups Data Set

Algorithm	Target Domain							
	os_versus_crypt		ibm_versus_electronics		mac_versus_med		x_versus_space	
	Mistake (%)	Time (s)	Mistake (%)	Time (s)	Mistake (%)	Time (s)	Mistake (%)	Time (s)
PA	16.00±0.41	0.13±0.01	24.43±0.82	0.20±0.05	15.60±0.41	0.22±0.03	15.20±0.55	0.20±0.04
PAIO	13.75±0.31	0.14±0.02	22.38±0.68	0.33±0.06	13.05±0.48	0.30±0.05	13.28±0.52	0.27±0.06
HomOTL-I	10.56±0.30	0.48±0.11	23.88±0.60	0.57±0.08	14.77±0.46	0.41±0.09	14.49±0.68	0.52±0.10
HomOTL-II	11.38±0.46	0.46±0.09	23.02±0.57	0.61±0.16	15.07±0.48	0.47±0.09	11.73±0.33	0.69±0.16
HomOTLMS	**8.18±0.28**	2.93±0.34	**20.16±0.65**	2.57±0.51	**12.86±0.67**	2.79±0.34	**9.17±0.34**	3.13±0.28

TABLE 6
Results (Mean±Standard Deviations) of Applying Different Learning Algorithms on the Sentiment Analysis Data Set

Algorithm	Target Domain							
	books		DVDs		electronics		kitchen	
	Mistake (%)	Time (s)	Mistake (%)	Time (s)	Mistake (%)	Time (s)	Mistake (%)	Time (s)
PA	48.05±0.29	0.22±0.03	48.25±0.58	0.23±0.03	48.98±0.37	0.13±0.01	50.05±0.06	0.09±0.01
PAIO	45.51±0.54	0.79±0.14	46.36±0.77	0.84±0.11	43.18±0.60	0.72±0.16	42.91±0.60	0.81±0.26
HomOTL-I	47.55±0.57	1.25±0.26	47.83±0.34	1.23±0.21	47.60±1.01	0.98±0.23	49.39±0.50	1.10±0.25
HomOTL-II	47.75±0.00	1.19±0.23	47.55±0.02	1.18±0.21	46.54±1.20	1.06±0.17	49.34±0.33	1.11±0.24
HomOTLMS	**43.64±1.34**	4.95±0.85	**45.23±0.68**	4.71±1.03	**42.91±0.82**	4.70±0.67	**39.68±0.48**	4.67±0.89

English documents, EN-EN for short), Index FR-EN(French documents translated to English, FR-EN for short), Index GR-EN (German documents translated to English, GR-EN for short), Index IT-EN (Italian documents translated to English, IT-EN for short) and Index SP-EN (Spanish documents translated to English, SP-EN for short). And similarly for the four other languages. For each language, there are six relatively populous categories: C15, CCAT, E21, ECAT, GCAT, and M11. In our experiment, we label the instances with respect to C15 as positive instances, and those instances of M11 are labeled negative. Thus we can construct five new data sets (called EN, FR, GR, IT, and SP, respectively) for each language, we choose the original file as target domain, and the remaining four translated files as source domains, e.g., for EN, EN-EN is chosen as target domain and FR-EN, GR-EN, IT-EN, SP-EN are chosen as source domains. Since the data set contains many discriminatory features (e.g., some high-frequency words), we use FGM (Feature Generating Machine) [35], [36] to pre-process the data to screen out discriminant features. Detailed information of five languages data set is summarized in Table 4.

6.2 Baselines

To evaluate the performance of the proposed algorithm, we compare our algorithm with several state-of-the-arts methods. Specifically, for the homogeneous situation: since HomOTLMS is an online learning algorithm, we compare HomOTLMS with Passive-Aggressive, which is a classical online learning algorithm [16]. Considering that Passive-Aggressive is not designed for the transfer learning problem, we implement a variant of PA algorithm which is denoted as "PAIO" for online transfer learning, by initializing PA with a classifier trained with the whole source domain. Also, we compare HomOTLMS with two well-known online transfer learning algorithms (HomOTL-I, HomOTL-II proposed in [31]). Considering that both HomOTL-I and HomOTL-II are single-source online transfer learning methods, we combined all the instances in different source domains as a single source domain for HomOTL-I and HomOTL-II.

For the heterogeneous situation: since HetOTLMS is designed for learning problems where source and target domain have heterogeneous feature spaces, we compare HetOTLMS with the well-known algorithms HetOTL [31] which is a heterogeneous online transfer learning method. HetOTL uses the same heterogeneous data generation setting in the experiment. Based on HetOTL, we implement a variant, called "HetOTL0", by running the HetOTL algorithm only using the part of the features in the data set. We also compare HetOTLMS with HetOTL0 and PAIO. For the heterogeneous situation, in PAIO, we run the PA

TABLE 7
Results (Mean±Standard Deviations) Using Different Homogeneous Learning Algorithms on *EN-EN*

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	MCC (%)
PA	86.55±0.19	91.26±0.14	92.19±0.17	91.73±0.12	55.53±0.83
PAIO	87.63±0.22	92.42±0.18	92.27±0.15	92.35±0.13	60.62±1.07
HomOTL-I	85.06±0.53	93.23±0.30	87.91±0.95	90.49±0.40	62.54±0.97
HomOTL-II	86.48±0.18	92.43±0.44	90.72±0.54	91.56±0.13	62.40±0.27
HomOTLMS	**89.52±0.12**	**94.20±0.08**	**92.75±0.11**	**93.47±0.08**	**66.91±0.29**
Confidence Interval	[89.46, 89.58]	[94.17, 94.24]	[92.70, 92.80]	[93.43, 93.51]	[66.77, 67.05]

TABLE 8
Results (Mean±Standard Deviations) Using Different Homogeneous Learning Algorithms on *FR-FR*

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	MCC (%)
PA	90.86±0.25	91.12±0.25	91.66±0.32	91.39±0.24	81.38±0.45
PAIO	91.74±0.26	92.11±0.24	92.30±0.34	92.20±0.25	83.19±0.33
HomOTL-I	93.87±0.01	**94.61±0.01**	93.75±0.01	94.18±0.01	88.14±0.01
HomOTL-II	93.87±0.00	**94.61±0.00**	93.76±0.00	94.18±0.00	**88.15±0.00**
HomOTLMS	**94.03±0.08**	92.31±0.11	**96.78±0.06**	**94.49±0.07**	88.04±0.13
Confidence Interval	[93.99, 94.06]	[92.25, 92.36]	[96.75, 96.81]	[94.46, 94.52]	[87.98, 88.10]

TABLE 9
Results (Mean±Standard Deviations) Using Different Homogeneous Learning Algorithms on *GR-GR*

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	MCC (%)
PA	81.99±0.37	82.03±0.35	81.93±0.47	81.98±0.38	64.04±0.78
PAIO	83.01±0.35	83.15±0.40	82.81±0.35	82.98±0.34	66.15±0.64
HomOTL-I	88.29±0.01	**91.88±0.01**	84.01±0.02	87.77±0.01	77.50±0.02
HomOTL-II	88.30±0.00	**91.88±0.01**	84.02±0.00	87.78±0.00	77.52±0.01
HomOTLMS	**89.30±0.09**	89.25±0.07	**89.37±0.21**	**89.31±0.10**	**78.76±0.11**
Confidence Interval	[89.26, 89.34]	[89.21, 89.28]	[89.27, 89.47]	[89.26, 89.35]	[78.71, 78.81]

TABLE 10
Results (Mean±Standard Deviations) Using Different Homogeneous Learning Algorithms on *IT-IT*

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	MCC (%)
PA	84.27±0.26	79.07±0.28	**87.52±0.41**	83.08±0.29	69.00±0.49
PAIO	84.22±0.27	79.06±0.30	87.40±0.39	83.02±0.30	68.79±0.49
HomOTL-I	86.55±0.07	83.18±0.10	87.14±0.09	85.11±0.08	72.82±0.11
HomOTL-II	86.46±0.00	82.93±0.01	87.30±0.00	85.06±0.00	72.67±0.00
HomOTLMS	**87.12±0.13**	**84.18±0.15**	87.21±0.30	**85.67±0.16**	**74.10±0.37**
Confidence Interval	[87.06, 87.18]	[84.11, 84.25]	[87.07, 87.35]	[85.59, 85.75]	[73.93, 74.28]

TABLE 11
Results (Mean±Standard Deviations) Using Different Homogeneous Learning Algorithms on *SP-SP*

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	MCC (%)
PA	90.03±0.26	61.81±1.16	58.65±0.94	60.19±0.94	54.23±1.18
PAIO	**91.40±0.31**	**65.66±1.22**	69.23±1.11	67.40±1.12	62.34±1.03
HomOTL-I	90.38±0.22	58.82±1.06	84.24±4.11	69.20±0.99	64.67±1.69
HomOTL-II	90.12±0.11	57.65±0.43	**86.97±2.81**	69.32±0.77	65.57±0.80
HomOTLMS	90.49±0.07	58.84±0.21	86.47±0.25	**70.03±0.17**	**66.27±0.20**
Confidence Interval	[90.46, 90.52]	[58.74, 58.94]	[86.36, 86.59]	[69.95, 70.11]	[66.18, 66.37]

algorithm using the first part of the features in the target domain by initializing with a classifier from the source domain. The code of the algorithm can be downloaded from the github.[2]

6.3 Experimental Results on 20Newsgroups and Sentiment Analysis Data Sets

We first compare HomOTLMS with the baselines on the 20Newsgroups and sentiment analysis data sets. Tables 5 and 6 show the performance of the algorithms on two data sets. According to the experimental result, the overall data

[2] https://github.com/wuhanrui/TKDE2016submission.git

TABLE 12
Characteristics of Data Used as Source Domains for Heterogeneous Online Transfer Learning

Domain	# Instances	# Dimensions
os_versus_crypt	1952	10,000
ibm_versus_electronics	1963	15,000
mac_versus_med	1945	20,000
x_versus_space	1967	25,000
books	2000	100,000
DVDs	2000	200,000
electronics	2000	300,000
kitchen	2000	400,000

taken from the experiments are clearly in favor of the proposed method. On these two data sets, HomOTLMS obtains the best performance over the baselines on all the tasks. The reason is that the baselines do not take into account the distribution difference between the source domains and target domain. Directly applying the classifier learned on on the source domains to the target domain may lead to a poor result. On the contrary, HomOTLMS considers the difference between the source domains and target domain and updates the classifier based on the loss of the target domain. By this way, HomOTLMS can effectively transfer knowledge from the source domains to target domain. In general, HomOTLMS can learn a better classifier for target domain and achieve better performance than baselines.

We also compare the efficiency of the proposed algorithm with baselines in terms of running time. The results are shown in Tables 5 and 6. Compared to HomOTL-I and HomOTL-II that are performed on a single source domain, HomOTLMS takes more running time because of the calculation on multiple source domains. Nevertheless, considering the better performance, the increased time cost is acceptable.

6.4 Experimental Results on Large Scale Data Set

For more in-depth understanding of the performance of our algorithm, we compare the proposed algorithm with baselines on a large scale data set, five language data set [34]. Tables 7, 8, 9, 10, and 11 show the performance of the algorithms in terms of accuracy, precision, recall, F-measure and Matthews Correlation Coefficient (MCC) [37]. To help understand the performances, we also show the confidence interval for each measure. For instance, if the accuracy of a baseline is less than the lower limit of the confidence interval, then our algorithm is better than the baseline; if the accuracy of a baseline falls into the confidence interval, then performance of this baseline is comparable to our algorithm; For other cases, our algorithm is less effective than the baseline. From Tables 7, 8, 9, 10, and 11, we can observe that HomOTLMS is more effective than the baselines on most of the tasks under the five measures in general. This is because the baselines use all the sources indiscriminately in the training process of the classification model. And, this practice ignores the correlation between each source domain and target domain. Different from the baselines, HomOTLMS trains classification model on each source domain and utilizes an ensemble-strategy for prediction. By this way, HomOTLMS can leverage the benefits from the source domain that is more relevant to the target domain, and avoid the adverse effects of the source domain that is less relevant to the target domain. Although HomOTLMS can achieve a better performance than baselines, even on a large scale data set, the improvements on the five language data set are marginal for some tasks. On a large scale data set, the size of target domain becomes larger than the data set used in the previous section, leading to the marginal improvements as transfer learning works better in case that the labeled data in target domain for training are insufficient.

6.5 Experimental Results on Heterogeneous Setting

In this experiment, we test the performance of HetOTLMS with heterogeneous settings. To generate the heterogeneous data sets where the feature set of the source domain is the subset of that of the target domain, we split the features of the data instances in each domain into two parts. For a source domain data set, we keep the first part and remove the second part. For a target domain, we use all the features in the data set. Table 12 shows the characteristics of the data used as source domains. The way we generate heterogeneous data sets is reasonable. Since we may encounter a similar situation in which only part of features can be obtained from one view in the real world. For instance, in a cross-view object recognition, the same object appears quite differently when different views are observed, classification models learned from one view may degrade the performance in another view. In such case, each view can be treated as a source domain, and the feature in each domain is different from the other domain. Note that we cannot directly combine the instances in multiple sources into a single source domain because of the dimension divergence of different sources. To apply these algorithms to the multiple sources data sets, we run the baseline method on the data set containing the target domain data and one source domain data. Results with different source domain data are recorded, and the best one is recorded.

Tables 13 and 14 show the results of different algorithms on the 20Newsgroups data set and the sentiment analysis data set, respectively. The proposed HetOTLMS algorithm

TABLE 13
Results of Different Heterogeneous Learning Algorithms on the 20Newsgroups Dataset

Algorithm	Target Domain							
	os_versus_crypt		ibm_versus_electronics		mac_versus_med		x_versus_space	
	Mistake (%)	Time (ms)	Mistake (%)	Time (ms)	Mistake (%)	Time (ms)	Mistake (%)	Time (ms)
PAIO	14.76±0.53	1054.5±6.6	24.15±0.35	1060.2±5.5	17.87±0.48	1048.1±5.4	16.17±0.38	1060.6±4.5
HetOTL0	22.52±0.39	736.2±3.5	19.08±0.34	722.2±3.7	18.96±0.35	711.8±2.8	18.38±0.25	718.3±2.5
HetOTL	22.16±0.39	1609.8±46	**18.81±0.28**	1.465.1±12	18.80±0.30	1451.1±21	18.02±0.22	1471.7±20.9
HetOTLMS	**12.65±0.58**	1865.1±6.6	18.87±0.75	1947.1±3.8	**13.60±0.50**	1916.2±6.2	**11.09±0.49**	2068.3±39.4

TABLE 14
Results of Different Heterogeneous Learning Algorithms on the Sentiment Analysis Dataset

Algorithm	Target Domain							
	books		DVDs		electronics		kitchen	
	Mistake (%)	Time (ms)	Mistake (%)	Time (ms)	Mistake (%)	Time (ms)	Mistake (%)	Time (ms)
PAIO	35.49±0.74	642.1±35.2	35.29±0.51	635.3±29.6	28.29±0.37	623.0±21.9	27.92±0.46	627.5±25.7
HetOTL0	34.13±0.37	558.0±26.9	34.69±0.51	557.6±26.2	30.89±0.42	547.6±17.4	28.92±0.41	555.3±25.7
HetOTL	33.28±0.30	913.2±46.9	33.83±0.47	908.9±47.0	30.27±0.39	896.4±31.3	27.89±0.39	902.0±38.5
HetOTLMS	**31.61±0.49**	1130.9±105	**33.30±0.72**	1114.7±77	**26.99±0.50**	1106.7±66	**23.52±0.60**	1115.7±108

consistently performs best, or close to the best, regarding mistake rate. The result demonstrates the effectiveness of the proposed method for heterogeneous online transfer learning. In addition, the time cost of different algorithms is also given in Tables 13 and 14. We can see that the running time of HetOTLMS is greater than that of the other algorithms. However, the algorithm can still be learned very fast.

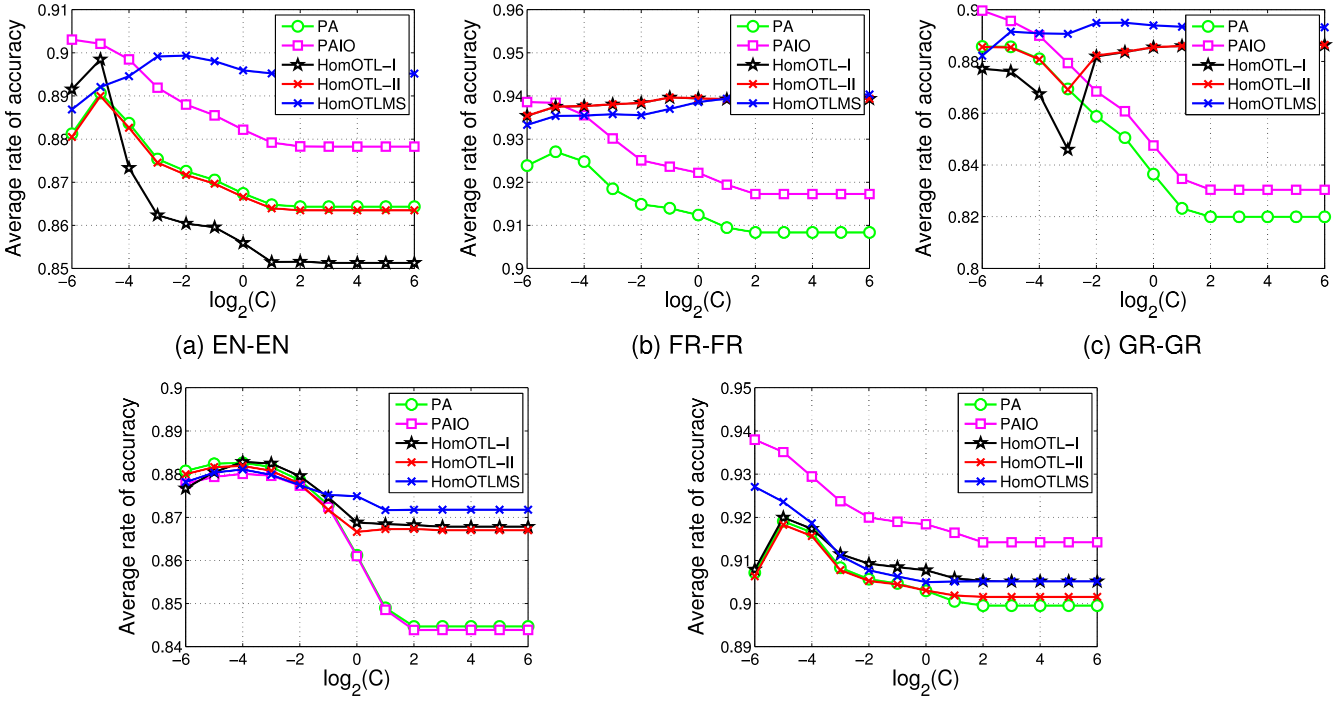
We also conduct experiments by varying number of learning samples in the target domain and present the results of different algorithms on the 20Newsgroups data set and the sentiment analysis data set in Figs. 7 and 8, respectively. From the result, the mistake rates of HetOTLMS are higher than that of the compared algorithm when the target domain contains a limited number of instances. The mistake rate of the HetOTLMS method decreases very rapidly when the number of instances increases, while the HetOTLMS method achieves better performance against the other algorithms when the number of instances in the target domain is sufficient (e.g., with 1,500 or 2,000 instances in the target domain). These results indicate that HetOTLMS can more effectively handle the online transfer learning problem across heterogeneous domains, where the source domains and the target domain have different feature spaces.

6.6 Parameter Tuning

The method proposed in this paper involves some tunable parameters, including the tradeoff parameter C. Fig. 6 reports the potential impacts imposed by different C on the five language data set. From the figure, we can observe that the accuracies of HomOTLMS and other baselines change significantly with different C. For the same task, different algorithms achieve their best performance on different values of C. For instance, in Fig. 6a, the best C for PA is 2^{-5} while the best C for HomOTLMS is 2^{-3}. We can draw a conclusion that HomOTLMS is more accurate than the other transfer learning strategies under varied C values, which validates the efficacy of the proposed online transfer learning strategy. In the experiments, we set C to be 5 for all the algorithms including HomOTLMS and the compared baselines.

6.7 Time Cost

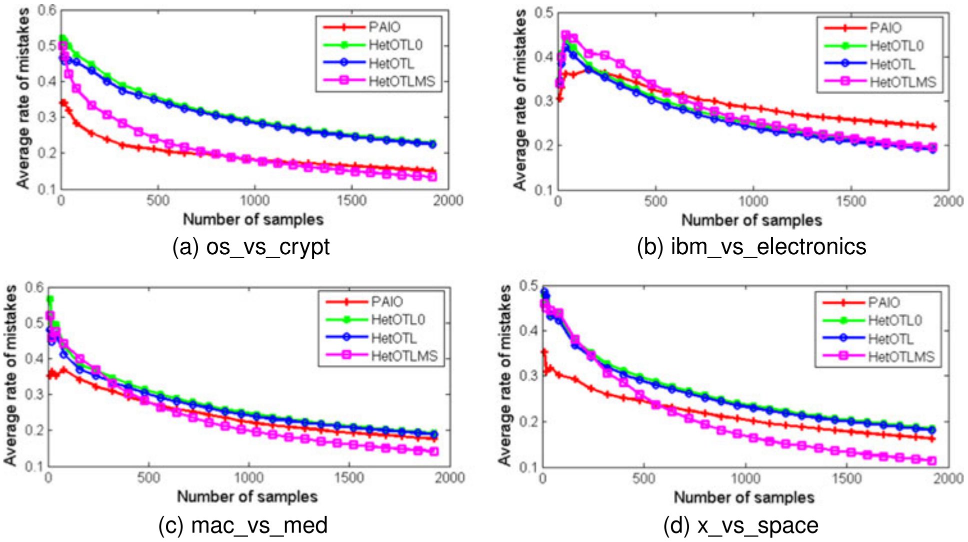
To evaluate the efficiency of our algorithm when using more source domains and/or more training instances, we test our algorithm on several tasks where the source number and the instance number in each source are both different. The experiments were implemented in MATLAB R2015b



(a) EN-EN (b) FR-FR (c) GR-GR

(d) IT-IT (e) SP-SP

Fig. 6. Evaluation on homogeneous OTL classification tasks with varied C values on five languages dataset.



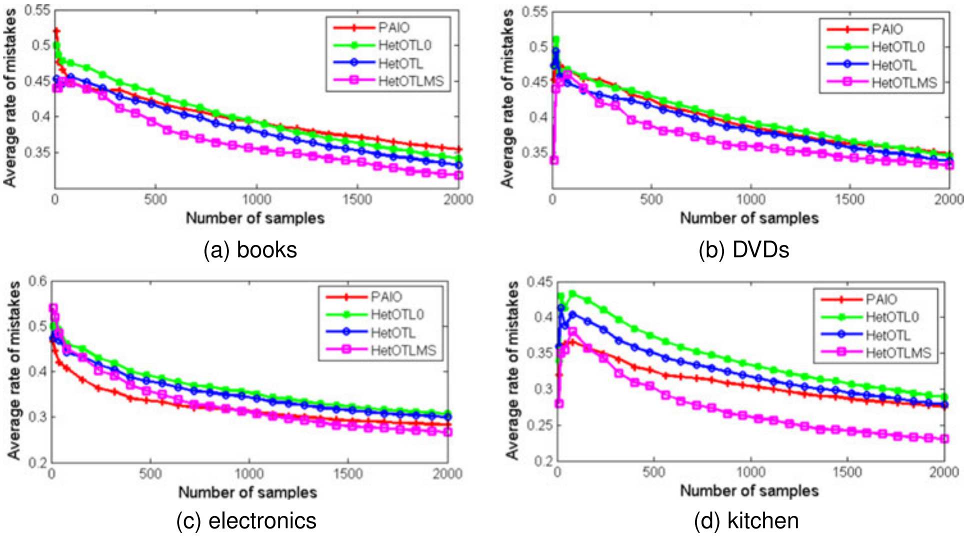
(a) os_vs_crypt

(b) ibm_vs_electronics

(c) mac_vs_med

(d) x_vs_space

Fig. 7. Average mistake rates of different heterogeneous learning algorithms with the increase of target domain instances on 20Newsgroups dataset.



(a) books

(b) DVDs

(c) electronics

(d) kitchen

Fig. 8. Average mistake rates of different heterogeneous learning algorithms with the increase of target domain instances on sentiment analysis dataset.

and run on a Windows machine with 2 × 2.4 GHz CPU processors (Intel(R) Xeon(R)) and 128 GB memory. The average running time of our algorithm is recorded and summarized in Fig. 9. From the figure, we can find that the average runtime of our algorithm spends more and more as the number of sources and the number of instances increases. We also notice that, for an identical increment instance number, the time cost for the task with more sources increases faster than the task with fewer sources.

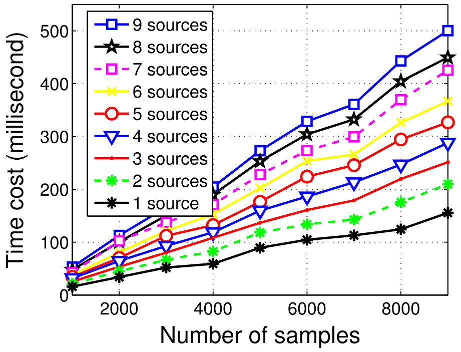


Fig. 9. Time cost as the number of samples increases.

7 Conclusion

In this paper, we propose an online transfer learning paradigm with multiple source domains. A homogenous online transfer learning algorithm and a heterogeneous online transfer learning algorithm are both designed. Compared to the traditional online transfer learning algorithms which were only designed for learning from a single source domain, our methods can transfer knowledge from multisource domains, even the source domains that have heterogeneous feature space with the target domain. Based on the theoretical analysis, we can have a clearer understanding of the parameters of the algorithm on the mistake. Experimental results on three real-world data sets indicate that the proposed algorithms are able to achieve better performances against the compared baselines, and the time cost increases are acceptable.

Several problems remain to be investigated in our future work. First, we only propose algorithms to tackle binary classification tasks. Thus, a multi-class classification problem is one way to extend our proposed algorithms. Second, in our heterogeneous settings, features in the source domain constitute a subset of those of the target domain. This may limit the application scenario for our algorithm. Hence, extending our algorithm settings to a more general situation is of another future work. Lastly, due to the large size of the target domain on a large scale data set, the improvements become marginal. In the future, we will use some techniques w.r.t. transitive transfer learning, such as source domain selection or instances selection, to achieve more improvements.

Acknowledgments

Q. Wu and H. Wu are co-first authors. Y. Yan and T. Hao are the corresponding authors. This work was supported by the Guangzhou Key Laboratory of Robotics and Intelligent Software under Grant 15180007, and the National Natural Science Foundation of China (NSFC) under Grants 61502177, 61403088 and 61602185.

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] 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.

[3] E. W. Xiang, S. J. Pan, W. Pan, J. Su, and Q. Yang, "Source-selection-free transfer learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, vol. 22, no. 3, Art. no. 2355.

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

[5] L. Ge, J. Gao, and A. Zhang, "OMS-TL: A framework of online multiple source transfer learning," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 2423–2428.

[6] 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.

[7] H. Xia, S. C. Hoi, R. Jin, and P. Zhao, "Online multiple kernel similarity learning for visual search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 536–549, Mar. 2014.

[8] J. Wang, S. C. Hoi, P. Zhao, and Z.-Y. Liu, "Online multi-task collaborative filtering for on-the-fly recommender systems," in *Proc. 7th ACM Conf. Recommender Syst.*, 2013, pp. 237–244.

[9] K. Singer, "Online classification on a budget," in *Proc. Advances Neural Inf. Process. Syst.*, 2004, vol. 16, Art. no. 225.

[10] S. Shalev-Shwartz and Y. Singer, "Online learning: Theory, algorithms, and applications," 2007.

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

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

[13] Y. Li and P. M. Long, "The relaxed online maximum margin algorithm," *Mach. Learn.*, vol. 46, no. 1–3, pp. 361–387, 2002.

[14] C. Gentile, "A new approximate maximal margin classification algorithm," *J. Mach. Learn. Res.*, vol. 2, pp. 213–242, 2001.

[15] K. Crammer and Y. Singer, "Ultraconservative online algorithms for multiclass problems," *J. Mach. Learn. Res.*, vol. 3, pp. 951–991, 2003.

[16] 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, 2006.

[17] P. Zhao, S. C. Hoi, and R. Jin, "Double updating online learning," *J. Mach. Learn. Res.*, vol. 12, pp. 1587–1615, 2011.

[18] M. Dredze, K. Crammer, and F. Pereira, "Confidence-weighted linear classification," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 264–271.

[19] J. Wang, P. Zhao, and S. C. Hoi, "Exact soft confidence-weighted learning," in *Proc. 29th Int. Conf. Mach. Learn. (ICML-12)*, 2012, pp. 121–128.

[20] Z. Deng, Y. Jiang, F. L. Chung, H. Ishibuchi, K.-S. Choi, and S. Wang, "Transfer prototype-based fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 5, pp. 1210–1232, Oct. 2016.

[21] F. Zhuang, P. Luo, C. Du, Q. He, Z. Shi, and H. Xiong, "Triplex transfer learning: Exploiting both shared and distinct concepts for text classification," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1191–1203, Jul. 2014.

[22] 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.

[23] M. Long, J. Wang, G. Ding, S. J. Pan, and P. S. Yu, "Adaptation regularization: A general framework for transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1076–1089, May 2014.

[24] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 465–479, Mar. 2012.

[25] B. Tan, Y. Song, E. Zhong, and Q. Yang, "Transitive transfer learning," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1155–1164.

[26] E. Eaton and M. desJardins, "Selective transfer between learning tasks using task-based boosting," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 337–342.

[27] Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1855–1862.

[28] B. Tan, E. Zhong, E. W. Xiang, and Q. Yang, "Multi-transfer: Transfer learning with multiple views and multiple sources," *Statist. Anal. Data Mining*, vol. 7, no. 4, pp. 282–293, 2014.

[29] M. Dredze, A. Kulesza, and K. Crammer, "Multi-domain learning by confidence-weighted parameter combination," *Mach. Learn.*, vol. 79, no. 1/2, pp. 123–149, 2010.

[30] Y. Jiang, F. L. Chung, H. Ishibuchi, Z. Deng, and S. Wang, "Multitask TSK fuzzy system modeling by mining intertask common hidden structure," *IEEE Trans. Cybern.*, vol. 45, no. 3, pp. 534–547, Mar. 2015.

[31] P. Zhao and S. C. Hoi, "OTL: A framework of online transfer learning," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 1231–1238.

[32] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Computational Learning Theory*. Berlin, Germany: Springer, 2001, pp. 416–426.

[33] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Proc. Eur. Conf. Computat. Learn. Theory*, 1995, pp. 23–37.

[34] M. Amini, N. Usunier, and C. Goutte, "Learning from multiple partially observed views-an application to multilingual text categorization," in *Proc. 22nd Int. Conf. Advances Neural Inf. Process. Syst.*, 2009, pp. 28–36.

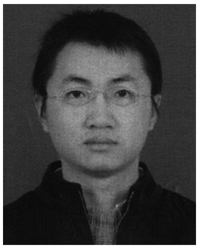
[35] M. Tan, L. Wang, and I. W. Tsang, "Learning sparse SVM for feature selection on very high dimensional datasets," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 1047–1054.

[36] M. Tan, I. W. Tsang, and L. Wang, "Towards ultrahigh dimensional feature selection for big data," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1371–1429, 2014.

[37] B. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochimica et Biophysica Acta-Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.

Qingyao Wu received the PhD degree in computer science from the Harbin Institute of Technology, in 2013. He is an associate professor in the School of Software Engineering, South China University of Technology. He worked as a postdoc research fellow in the School of Computer Engineering, Nanyang Technological University until to 2015. His research interests include machine learning, data mining, big data research, computer vision, and bioinformatics.

Hanrui Wu received the BS degree in software engineering from the South China University of Technology, China, in 2013. He is working toward the PhD degree in the School of Software Engineering, South China University of Technology, China. His research interests include transfer learning, online learning, and multi-label classification.



Xiaoming Zhou received the bachelor's degree from the School of Software, East China Normal University, Shanghai, China, and the master's degree from the School of Computer Science & Engineering, South China University of Technology, Guangzhou, China. His research interests include transfer learning, online learning, and data mining.

Mingkui Tan received the PhD degree in computer science from Nanyang Technological University, Singapore, in 2014. He is a professor in the School of Software Engineering, South China University of Technology. After that, he worked as a senior research associate in the School of Computer Science, University of Adelaide, Australia. His research interests include compressive sensing, big data learning, and large-scale optimization.

Yonghui Xu received the BS degree in information and computing science from Henan University, China, in 2011. He is working toward the PhD degree in the School of Software Engineering, South China University of Technology, China. His research interests include transfer learning, metric learning, and multi-instance multi-label learning and their applications in computer vision and bioinformatics engineering.

Yuguang Yan received the BS degree in software engineering from the South China University of Technology, China, in 2013. He is working toward the PhD degree in the School of Software Engineering, South China University of Technology, China. His research interests include transfer learning, multi-label classification, and online learning.

Tianyong Hao received the PhD degree in computer science from the City University of Hong Kong, in 2010. He is an associate professor with the Guangdong University of Foreign Studies. He visited York University, Canada, in 2008 and Emory University, from 2009 to 2010. After that, he worked as a research fellow with the City University of Hong Kong from 2010 to 2012. He further worked as a postdoc research scientist with Columbia University until 2014. His research interests include natural language processing, question answering, and clinical research informatics.