Heterogeneous Transfer Learning with Weighted Instance-Correspondence Data

Yuwei He,[1] Xiaoming Jin,[1]* Guiguang Ding,[1] Yuchen Guo,[1,2]*
Jungong Han,[3] Jiyong Zhang,[4] Sicheng Zhao[5]
[1]Beijing National Research Center for Information Science and Technology (BNRist)
[1]School of Software, [2]Department of Automation, Tsinghua University, Beijing 100084, China
[3]WMG Data Science, University of Warwick, Coventry, UK
[4]School of Automation, Hangzhou Dianzi University, China
[5]Department of EECS, University of California, Berkeley, USA
hyw16@mails.tsinghua.edu.cn, {xmjin, dinggg}@tsinghua.edu.cn,
{yuchen.w.guo, jungonghan77, schzhao}@gmail.com, jzhang@hdu.edu.cn

Abstract

Instance-correspondence (IC) data are potent resources for *heterogeneous* transfer learning (HeTL) due to the capability of bridging the source and the target domains at the instance-level. To this end, people tend to use machine-generated IC data, because manually establishing IC data is expensive and primitive. However, existing IC data machine generators are not perfect and always produce the data that are not of high quality, thus hampering the performance of domain adaption. In this paper, instead of improving the IC data generator, which might not be an optimal way, we accept the fact that data quality variation does exist but find a better way to use the data. Specifically, we propose a novel *heterogeneous* transfer learning method named *Transfer Learning with Weighted Correspondence* (TLWC), which utilizes IC data to adapt the source domain to the target domain. Rather than treating IC data equally, TLWC can assign solid weights to each IC data pair depending on the quality of the data. We conduct extensive experiments on HeTL datasets and the state-of-the-art results verify the effectiveness of TLWC.

Introduction

Transfer learning or domain adaptation builds machine learners that can be generalized across different domains, and is capable of leveraging rich label information from the source domain to the target domain. Transfer learning can effectively ease the manual effort of labeling data for the target domain (Pan and Yang 2010).

Different from *homogeneous* transfer learning (HoTL), where the source and the target domain share the same feature space, *heterogeneous* transfer learning (HeTL) allows the data in two domains to be represented with different feature spaces (Weiss, Khoshgoftaar, and Wang 2016). Therefore, how to unify the feature spaces of the two domains is a bottleneck of HeTL.

For HeTL, instance-correspondence (IC) data are valuable mediums to unify the feature spaces (i.e. a document presented both in English and French) (Zhu et al. 2011; Zhou et al. 2014). These data can be utilized to either establish a common feature space (Xiao and Guo 2013; Zhou et al. 2015; 2016) for the two domains or learn a feature transformation that maps data from one domain to another (Zhou et al. 2014; Guntuku et al. 2016; Wang, Cui, and Zhu 2018). Existing HeTL methods unify the two domains by matching marginal distributions at the domain-level or conditional distributions at the category-level. Provided with IC data, HeTL methods can undoubtedly unify domains better because they can bridge two domains at the instance-level.

Manually establishing IC data is primitive and expensive. Therefore, machine-generated data can be a substitute. Taking Natural Language Processing (NLP) tasks as examples, the IC documents can be generated with Machine

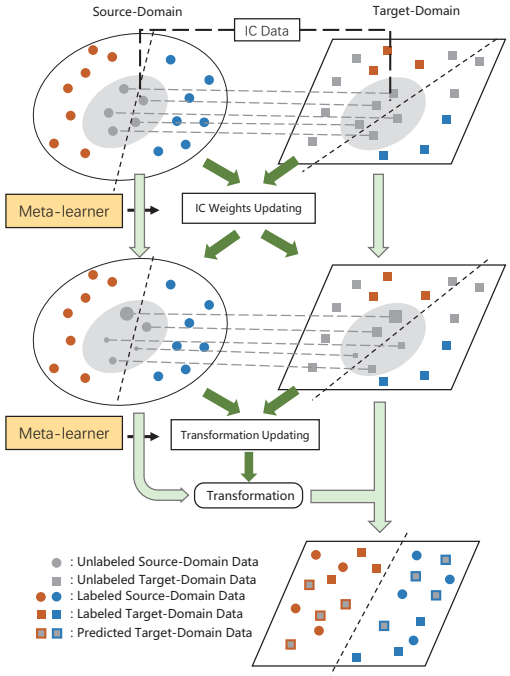


Figure 1: Framework of TLWC. IC data points sizes are proportional to their weights.

*Corresponding Author

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Translation. However, the qualities of machine-generated data sometimes fail to meet the requirement. For example, machine-translated sentences sometimes cannot accurately express the meaning of the original sentences and even cannot be sentenced. Bridging two domains with such rough IC data cannot achieve satisfactory results. Moreover, the quality of data should be judged by a floating score rather than just good or bad. Although there are existing instance weighting strategies, such as reweighting regularizer (Long et al. 2014) and kernel mean matching (Chu, La Torre, and Cohn 2013), they are not designed to reweight IC data and can only be applied for HoTL. Some coping strategies were designed, such as selecting high-quality correspondence with active learning (Zhou et al. 2016). However, they fail to assign precise weights to IC data and waste massive information.

To assign solid weights to IC data and learn more robust transformed features for both domains, in this paper, we propose a novel HeTL method named *Transfer Learning with Weighted Correspondence* (TLWC). Following a meta-learning paradigm (Finn, Abbeel, and Levine 2017; Ren et al. 2018), TLWC establishes a meta-learner to optimize the weights of IC data. The meta-objective of the meta-learner is to minimize the classification loss of limited labeled target-domain data. After the optimization is finished, weighted IC data are utilized to learn feature transformations that map the source-domain data to the target-domain. Then the transformations can be optimized by the meta-learner further. After the optimization procedure is finished, the classifier in the meta-learner is the one we need. The assumption we base is: If the IC data are assigned with the optimal weights, the feature transformations learned with them is most adaptive for the classification task in the target domain. The framework of TLWC is shown in Figure 1.

The contributions of the paper are summarized as follows: (1) We propose a novel *heterogeneous* transfer learning method, where the feature spaces of the two domains are unified with weighted IC data. Our method can be applied to different features, datasets, and modalities. (2) We are among the first to consider the IC data weights and we establish a meta-learner that utilizes the classification loss of target-domain data to guide the weights updating and transformation optimization procedure. Compared with previous HeTL methods based on IC data, TLWC can effective remove noisy information in IC data and learn a unified feature space which is more suitable for the classification task in the target domain.

Related Work

Transfer learning or domain adaptation aims at transferring rich label information from a source domain to a target domain. The main problem of transfer learning is how to reduce the data discrepancy between the two domains (Pan and Yang 2010; Weiss, Khoshgoftaar, and Wang 2016).

Depending on whether the feature spaces of the two domains are the same, transfer learning is divided into two categories: *homogeneous* transfer learning (HoTL) and *heterogeneous* transfer learning (HeTL). Different from HoTL, where the domain shift can be reduced by directly minimizing the data distribution discrepancy (Tzeng et al. 2015; Long et al. 2017), HeTL must unify the feature spaces of the two domains. For HeTL, there are two groups of methods for feature unification: (1) Projecting data from the two domains into a common feature space (Wang and Mahadevan 2011; Hoffman et al. 2014; Yang et al. 2015; Herath, Harandi, and Porikli 2017); (2) Mapping data from one domain to another one (Li et al. 2013; Hoffman et al. 2014; Xiao and Guo 2015; Tsai, Yeh, and Wang 2016).

For the first group, for example, Li *et al.* (2013) proposed a semi-supervised heterogeneous domain adaptation (SFHA), which transforms data from two domains into an augmented feature space. Herath *et al.* (2017) considered an invariant latent space (ILS) and they established a common Hilbert space for both domains.

For the second group, for example, Tsai *et al.* (2016) mapped the labeled source-domain data to the dimension-reduced target domain. And the mapped data are assigned with different weights. Hoffman *et al.* (2014) proposed a max-margin domain transformation (MMDT). With MMDT, the data in the target domain are projected to the source domain, while the projected data are classified by maximizing the margins.

The above HeTL methods bridge two domains by matching either the marginal or conditional distributions at the domain-level and the category-level respectively. With instance-correspondence (IC) data, the two domains can be bridged at the instance-level, which will boost the unification and adaptation of the two *heterogeneous* domains. According to the above two groups of feature unification methods, IC data can be utilized in various ways. For the first group of methods, which learns a common feature space for the two domains, IC data can be taken as the equivalent data in the common feature space. For example, Xiao and Guo (2013) proposed a two-step learning method which turns feature space learning into a matrix completion problem. The combined IC data are complete data in the matrix and they are employed to complete the unobserved data in the matrix. Zhou *et al.* (2015) designed a subspace for both domains and the projected IC data in the subspace are the same. For the second group of methods, which maps data from one domain to another, IC data are applied to learn the feature transformations. For example, Zhou *et al.* (2014) proposed a hybrid heterogeneous transfer learning (HHTL), which learns several transformation matrices with high-level features of IC data. Wang *et al.* (2018) applied deep autoencoder to embed the shallow representations of both domains and learned a feature transformation with IC data features at the top embedding layer.

However, all of the above methods consider the IC data to be equally important, which is not reliable. The IC data with low quality will hamper the domain adaptation procedure. Although there are existing instance weighting strategies (Chu, La Torre, and Cohn 2013; Long et al. 2014), they are only designed for HoTL. Zhou *et al.* (2016) selectively labeled IC data with active learning, but manually labeling IC data is prohibitive and informative machine-generated data are given up.

We assigned different weights to all the IC data pairs and optimized these weights as well as feature transformation with meta-learning. Meta-learning aims at optimizing parameter or hyper-parameter of a learner by establishing a high-level learner. The high-level learner can be divided into two groups: 1. Learner based on hyper-network (Andrychowicz et al. 2016; Snell, Swersky, and Zemel 2017; Jin et al. 2018); 2 Learner based on gradient descent (Finn, Abbeel, and Levine 2017; Ren et al. 2018). For the first group, a hyper-network, such as LSTM, is constructed to guide the parameter updating of the original leaner. For the second group, a meta-objective is established, which is derivable to the parameters to be updated. The meta-objective can be validation loss of the original learner. In the beginning, meta-learning aims to supervise the updating procedure of the original learner and guide it to learn to learn better (Andrychowicz et al. 2016; Finn, Abbeel, and Levine 2017). Recently, instance weights are also considered to be updated with a meta-learner (Jin et al. 2018; Ren et al. 2018).

Proposed Method

Problem Formulation

Our problem is to establish a classifier for a target domain with limited labeled data. To achieve this goal, we need to transfer knowledge from a source domain with rich label information to the target domain. The given data are: sufficient labeled data $\{\mathbf{X}^{S,l}, \mathbf{y}^{S,l}\} = \{(\mathbf{x}_i^{S,l}, y_i^S)\}_{i=1}^{n_{S,l}}$ and unlabeled data $\{\mathbf{X}^{S,u}\} = \{\mathbf{x}_i^{S,u}\}_{i=1}^{n_{S,u}}$ in the source domain; a set of labeled data $\{\mathbf{X}^{T,l}, \mathbf{y}^{T,l}\} = \{(\mathbf{x}_i^{T,l}, y_i^T)\}_{i=1}^{n_{T,l}}$ and unlabeled data $\{\mathbf{X}^{T,u}\} = \{\mathbf{x}_i^{T,u}\}_{i=1}^{n_{T,u}}$ in the target domain, where $n_{S,l} \gg n_{T,l}$; a set of unlabeled IC data $\{\mathbf{X}^{S,c}, \mathbf{X}^{T,c}\} = \{(\mathbf{x}_i^{S,c}, \mathbf{x}_i^{T,c})\}_{i=1}^{n_c}$ across the two domains. $\mathbf{x}_i^{S,l}$, $\mathbf{x}_i^{S,u}$ and $\mathbf{x}_i^{S,c}$ are in $\mathbb{R}^{d_S}$ while $\mathbf{x}_i^{T,l}$, $\mathbf{x}_i^{T,u}$ and $\mathbf{x}_i^{T,c}$ are in $\mathbb{R}^{d_T}$, where $d_s \neq d_T$. $n_S = n_{S,l} + n_{S,u} + n_c$ and $n_T = n_{T,l} + n_{T,u} + n_c$.

To solve the problem, we propose a HeTL method named *Transfer Learning with Weighted Correspondence* (TLWC), which consists of two components: (1) High-level feature learning; (2) IC data weights and transformation updating. For the first component, we learn high-level features for both domains with a marginalized stacked denoised autoencoder (mSDA). For the second component, we establish a meta-learner to optimize the weights of IC data and utilize weighted IC data to learn feature transformations to bridge two domains. Then the feature transformations are updated further by the meta-learner. When the optimization procedure is finished, the classifier for the target domain in the meta-learner is what we need.

High-Level Feature Learning

We employ a marginalized stacked denoised autoencoder (mSDA) on both to learn high-level features (Chen et al. 2012; Zhou et al. 2014). mSDA simplifies SDA from feature reconstruction to feature mapping, which makes the computation much more efficient.

Denoting $* \in \{S, T\}$, to learn the feature mapping matrix $\mathbf{W}^* \in \mathbb{R}^{d_* \times (d_*+1)}$, which is used to generate high-level features, we minimize the overall squared loss:

$$\mathcal{L}_{sq}(\mathbf{W}) = \sum_{i=1}^{n_*} \left\| \mathbf{x}_i^* - \mathbf{W}^* [\widetilde{\mathbf{x}}_i^*; 1] \right\|^2. \quad (1)$$

Each dimension of $\mathbf{x}_i^*$ is corrupted to 0 with a probability p, and $\widetilde{\mathbf{x}}_i^*$ is the expected version of corrupted $\mathbf{x}_i^*$. To incorporated the bias term, we absorb 1 into $\widetilde{\mathbf{x}}_i^*$. Defining a surviving feature vector $\mathbf{q} = [1-p, ..., 1-p, 1] \in \mathbf{R}^{d_*+1}$ and $\bar{\mathbf{X}}^* = [\mathbf{X}^*; \mathbf{1}^\top] \in \mathbb{R}^{(d_*+1)\times n_*}$, we can write the analytic solution of $\mathbf{W}^*$ as:

$$\mathbf{W}^* = \mathbf{P}\mathbf{Q}^{-1}, \quad (2)$$

where $\mathbf{P}_{ij} = \left(\mathbf{X}^* \bar{\mathbf{X}}^{*\top}\right)_{ij} \mathbf{q}_j$, and

$$\mathbf{Q}_{ij} = \begin{cases} \left(\overline{\mathbf{X}}^* \overline{\mathbf{X}}^{*\top}\right)_{ij} \mathbf{q}_i \mathbf{q}_j, & \text{if } i \neq j \\ \left(\overline{\mathbf{X}}^* \overline{\mathbf{X}}^{*\top}\right)_{ij} \mathbf{q}_i, & \text{otherwise} \end{cases} \quad (3)$$

With the reconstruction weights $\mathbf{W}^*$, we can generate nonlinear features $\mathbf{H}^*$, where $\mathbf{H}^* = \tanh(\mathbf{W}^* \bar{\mathbf{X}}^*)$. After recursively doing the procedure by replacing $\bar{\mathbf{X}}^*$ with $\bar{\mathbf{H}}^*$ for $K-1$ times, we can obtain K layers of high-level features, where $\bar{\mathbf{H}}_k^*$ is the features at the k^{th} layer and $\bar{\mathbf{H}}_1^* = \bar{\mathbf{X}}^*$.

Weights and Transformation Updating

We define a weight vector $\boldsymbol{\epsilon} \in \mathbb{R}^{n^c}$ for IC data features, which can be applied to establish feature transformations $\{\mathbf{G}_k \in \mathbf{R}^{d_t \times (d_s+1)}\}_{k=1}^K$ for each layer of two domains. To optimize $\boldsymbol{\epsilon}$ and learn more solid feature transformations for domain adaptation, we construct a meta-learner for $\boldsymbol{\epsilon}$ and $\{\mathbf{G}_k\}_{k=1}^K$. The meta-objective of the learner consists of two terms: (1) Small classification loss on $\{\mathbf{X}^{T,l}, \mathbf{y}^{T,l}\}$. (2) Acceptable distribution discrepancy between the two domains. The meta-learner is a computation graph w.r.t. $\boldsymbol{\epsilon}$. We denote $\boldsymbol{\epsilon}$ as $\boldsymbol{\epsilon}_t$ at each training step and $\boldsymbol{\epsilon}_1 = \mathbf{1}$.

To establish the meta-learner, at each training step t, we first learn the feature transformation $\mathbf{G}_k$, which maps features from the source to the target domain at k^{th} layer. $\mathbf{G}_k$ is learned with IC data features $\mathbf{H}_k^{S,c}$, $\mathbf{H}_k^{T,c}$ and the objective function to be minimized is:

$$\mathcal{L}_{p,k}(\boldsymbol{\epsilon}_t, \mathbf{G}_k) = \|\mathbf{H}_k^{T,c} \boldsymbol{\Upsilon}_t - \mathbf{G}_k \bar{\mathbf{H}}_k^{S,c} \boldsymbol{\Upsilon}_t\|_F^2 + \lambda \|\mathbf{G}_k\|_F^2, \quad (4)$$

where $\bar{\mathbf{H}}_k^{S,c} = [\mathbf{H}_k^{S,c}; \mathbf{1}^\top] \in \mathbb{R}^{(d_S+1)\times n_c}$. $\boldsymbol{\epsilon}$ is a diagonal matrix and $\boldsymbol{\Upsilon}_t = \mathrm{diag}(\boldsymbol{\epsilon})$. λ is the parameter of the regularization term. The closed form solution for Eq. 4 can be written as:

$$\mathbf{G}_k(\boldsymbol{\epsilon}_t) = (\bar{\mathbf{H}}_k^{S,c} \boldsymbol{\Upsilon}_t \boldsymbol{\Upsilon}_t^\top \mathbf{H}_k^{T,c\top})(\mathbf{H}_k^{T,c} \boldsymbol{\Upsilon}_t \boldsymbol{\Upsilon}_t^\top \mathbf{H}_k^{T,c} + \lambda \mathbf{I})^{-1}. \quad (5)$$

To measure the classification loss on $\{\mathbf{X}^{T,l}, \mathbf{y}^{T,l}\}$, which is the first term of the meta-objective, we establish a classifier f_θ for the target domain, which requires the outputs to be derivable to the inputs, such as a logistic regression classifier or a neural network model. Let θ be the model

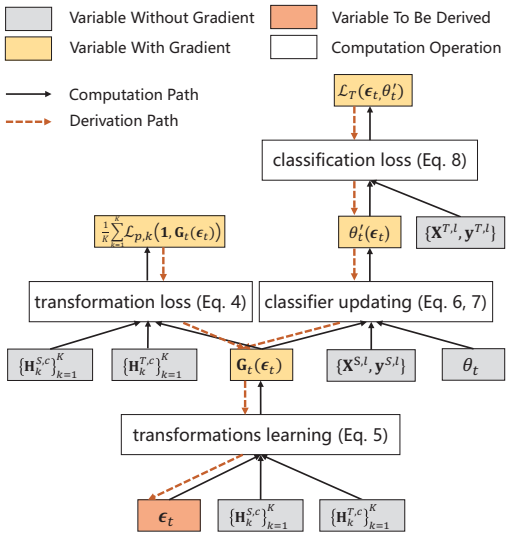


Figure 2: Computation graph for updating $\boldsymbol{\epsilon}$

.

parameters and $f(\mathbf{x}, y, \theta)$ be the classification loss for example $\mathbf{x}$. The inputs of f_θ are the concatenated high-level features in the target domain. For an instance $\mathbf{x}^T$, its features vector for f_θ is $\mathbf{z}^T = [\mathbf{h}_1^T; ...; \mathbf{h}_K^T] \in \mathbb{R}^{K \cdot d_T}$, where $\mathbf{h}_k^T$ is the corresponding features of $\mathbf{x}^T$ at k^{th} layer and $\mathbf{h}_1^T = \mathbf{x}^T$. And for an instance $\mathbf{x}^S$, its features vector for f_θ is $\mathbf{z}^S(\boldsymbol{\epsilon}_t) = [\mathbf{G}_1(\boldsymbol{\epsilon}_t)\mathbf{h}_1^S; ...; \mathbf{G}_K(\boldsymbol{\epsilon}_t)\mathbf{h}_K^S] \in \mathbb{R}^{K \cdot d_S}$. We denote $\mathbf{Z}_t^S = \{\mathbf{z}^S(\boldsymbol{\epsilon}_t)\}_{i=1}^{n_s}$ and $\mathbf{Z}^T = \{\mathbf{z}^T\}_{i=1}^{n_T}$.

At each training step t, we take a one-step batch gradient descent with $\{\mathbf{z}_i^{S,l}(\boldsymbol{\epsilon}_t), y_i^{S,l}\}_{i=1}^{n_{S,l}}$ to update θ_t:

$$\mathcal{L}_S(\boldsymbol{\epsilon}_t, \theta_t, \mathbf{Z}_t) = \frac{1}{n_{S,l}} \sum_{i=1}^{n_{S,l}} f\Big(\mathbf{z}_i^{S,l}(\boldsymbol{\epsilon}_t), y_i^{S,l}, \theta_t\Big), \quad (6)$$

$$\theta_t'(\boldsymbol{\epsilon}_t) = \theta_t - \alpha \nabla \mathcal{L}_S(\boldsymbol{\epsilon}_t, \theta_t, \mathbf{Z}_t). \quad (7)$$

Now the two terms of the meta-objective can be formed. The first term, which aims at minimizing the classification loss of $f_{\theta_t'}$ on $\{\mathbf{z}_i^{T,l}, y_i^{T,l}\}_{i=1}^{n_{T,l}}$, is written as:

$$\mathcal{L}_T(\boldsymbol{\epsilon}_t, \theta_t') = \frac{1}{n_{T,l}} \sum_{i=1}^{n_{T,l}} f\Big(\mathbf{z}_i^{T,l}, y_i^{T,l}, \theta_t'(\boldsymbol{\epsilon}_t)\Big). \quad (8)$$

The second term is the transformation loss in Eq. 4, which prevents $\boldsymbol{\epsilon}_t$ from overfitting to the limited labeled target-domain data. The two terms are utilized to update both IC weights $\boldsymbol{\epsilon}$ and transformations $\{\mathbf{G}_k\}_{k=1}^K$.

Data Weights Updating To optimize $\boldsymbol{\epsilon}_t$ with the meta-objective, we take a single gradient descent step w.r.t. $\boldsymbol{\epsilon}_t$:

$$\nabla \boldsymbol{\epsilon}_t = \frac{\partial}{\partial \boldsymbol{\epsilon}_t}\Big(\mathcal{L}_T(\boldsymbol{\epsilon}_t, \theta_t') + \eta \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_{p,k}(\mathbf{1}, \mathbf{G}_k(\boldsymbol{\epsilon}_t))\Big). \quad (9)$$

Before updating $\boldsymbol{\epsilon}_t$ with $\nabla \boldsymbol{\epsilon}_t$, we normalize the $\nabla(\boldsymbol{\epsilon}_t)$ with

$$\nabla \widetilde{\boldsymbol{\epsilon}}_t = \frac{\nabla \boldsymbol{\epsilon}_t}{\|\nabla \boldsymbol{\epsilon}_t\|_\infty + \delta(\|\nabla \boldsymbol{\epsilon}_t\|_\infty)}, \quad (10)$$

Algorithm 1 Transfer Learning with Weighted Correspondence

Require: $\{\mathbf{X}^{S,l}, \mathbf{y}^{S,l}\}$; $\{\mathbf{X}^{S,u}\}$; $\{\mathbf{X}^{T,l}, \mathbf{y}^{T,l}\}$; $\{\mathbf{X}^{T,u}\}$; $\{\mathbf{X}^{S,c}, \mathbf{X}^{T,c}\}$; number K of high-level layers. IC data weights vector $\boldsymbol{\epsilon}$; a classifier f_θ with parameter θ for the target domain; parameter $\lambda, \eta, \alpha, \gamma$.

Learning high-level features $\mathbf{H}_k^S$ and $\mathbf{H}_k^T$ for both domains
Initialize $\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon}_1 = \mathbf{1}$
for $t = 1$ to M, with step size 2, **do**
  $\mathbf{G}_k(\boldsymbol{\epsilon}_t) \leftarrow$ Eq. 5, $k = 1...K$
  $\mathcal{L}_S(\boldsymbol{\epsilon}_t, \theta_t) \leftarrow$ Eq. 6
  $\theta_t'(\boldsymbol{\epsilon}_t) = \theta_t - \alpha \nabla \mathcal{L}_S(\boldsymbol{\epsilon}_t, \theta_t, \mathbf{Z}_t)$
  $\mathcal{L}_T(\boldsymbol{\epsilon}_t, \theta_t') \leftarrow$ Eq. 8
  $\nabla \boldsymbol{\epsilon}_t \leftarrow$ Eq. 9; $\nabla \widetilde{\boldsymbol{\epsilon}}_t \leftarrow$ Eq. 10
  $\boldsymbol{\epsilon}_{t,i}' = \max(\boldsymbol{\epsilon}_{t,i} - \gamma \nabla \widetilde{\boldsymbol{\epsilon}}_{t,i}, 0)$
  $\boldsymbol{\epsilon}_{t+1} = \boldsymbol{\epsilon}_t' / \|\boldsymbol{\epsilon}_t'\|_\infty$
  $\theta_{t+1} = \theta_t - \alpha \nabla \mathcal{L}_S(\boldsymbol{\epsilon}_{t+1}, \theta_t)$
  $\nabla \mathbf{G}_k(\boldsymbol{\epsilon}_{t+1}) \leftarrow$ Eq. 14, k=1..K
  $\mathbf{G}_k'(\boldsymbol{\epsilon}_{t+1}) = \mathbf{G}_k(\boldsymbol{\epsilon}_{t+1}) - \beta \nabla \mathbf{G}_k(\boldsymbol{\epsilon}_{t+1})$, k = 1..K
  $\theta_{t+2} = \theta_{t+1} - \alpha \nabla \mathcal{L}_S(\boldsymbol{\epsilon}_{t+1}, \theta_{t+1}, \mathbf{Z}_{t+1}')$
end for
Fine-tune f_{θ_M} with $\{\mathbf{X}^{T,l}, \mathbf{y}^{T,l}\}$

Ensure: The classifier f_{θ_M} for the target domain.

where δ is to prevent denominator from being 0. $\delta(a) = 1$ if $a = 0$, and equals to 0 in other cases. Then $\boldsymbol{\epsilon}_t$ can be updated with:

$$\boldsymbol{\epsilon}_{t,i}' = \max(\boldsymbol{\epsilon}_{t,i} - \gamma \nabla \widetilde{\boldsymbol{\epsilon}}_{t,i}, 0), \quad (11)$$

$$\boldsymbol{\epsilon}_{t+1} = \boldsymbol{\epsilon}_t' / \|\boldsymbol{\epsilon}_t'\|_\infty. \quad (12)$$

Eq. 12 guarantees that $\|\boldsymbol{\epsilon}_{t+1}\|_\infty$ is no more than 1. The computation graph of updating $\boldsymbol{\epsilon}_t$ is shown in Figure 2. θ_t is simultaneously optimized with $\boldsymbol{\epsilon}_t$. Based on $\boldsymbol{\epsilon}_t$, θ_t is updated with:

$$\theta_{t+1} = \theta_t - \tau \nabla \mathcal{L}_S(\boldsymbol{\epsilon}_{t+1}, \theta_t, \mathbf{Z}_t). \quad (13)$$

Transformation Updating Firstly, with the optimized $\boldsymbol{\epsilon}_{t+1}$, we can update $\mathbf{G}_k(\boldsymbol{\epsilon}_t)$ to $\mathbf{G}_k(\boldsymbol{\epsilon}_{t+1})$ with Eq. 5. Then we utilize our meta-objective to update $\mathbf{G}_k(\boldsymbol{\epsilon}_{t+1})$ further:

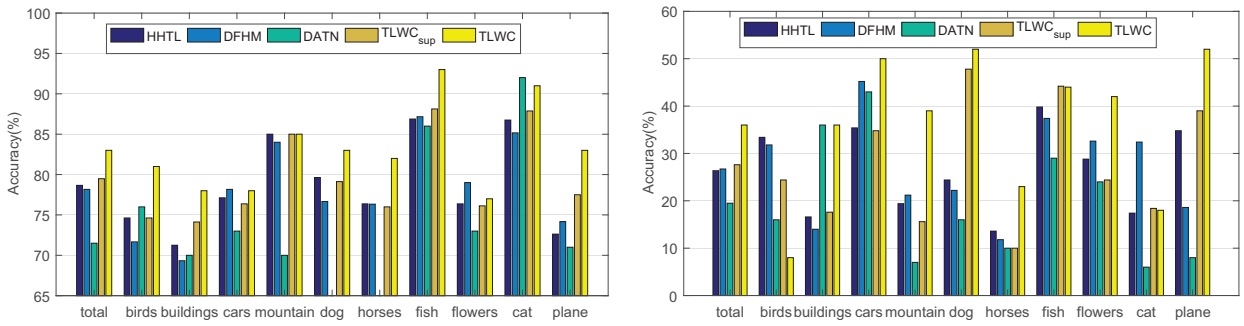
$$\nabla \mathbf{G}_k(\boldsymbol{\epsilon}_{t+1}) = \frac{\partial}{\mathbf{G}_k(\boldsymbol{\epsilon}_{t+1})} \big(\mathcal{L}_T(\boldsymbol{\epsilon}_{t+1}, \theta_{t+1}) + \eta \mathcal{L}_{p,k}(\mathbf{1}, \mathbf{G}_k(\boldsymbol{\epsilon}_{t+1}))\big) \quad (14)$$

$$\mathbf{G}_k'(\boldsymbol{\epsilon}_{t+1}) = \mathbf{G}_k(\boldsymbol{\epsilon}_{t+1}) - \beta \nabla \mathbf{G}_k(\boldsymbol{\epsilon}_{t+1}). \quad (15)$$

As the optimization of $\{\mathbf{G}_k\}_{k=1}^K$ is guided by the classification loss of the target domain, the high-level features $\mathbf{Z}_{t+1}'$ transformed by $\{\mathbf{G}_k'(\boldsymbol{\epsilon}_{t+1})\}_{k=1}^K$ become more suitable to update the classifier f_θ. The updating step is similar to Eq. 7.

$$\theta_{t+2} = \theta_{t+1} - \tau \nabla \mathcal{L}_S(\boldsymbol{\epsilon}_{t+1}, \theta_{t+1}, \mathbf{Z}_{t+1}'). \quad (16)$$

We repeat the above procedure for updating $\boldsymbol{\epsilon}$ and $\{\mathbf{G}_k\}_{k=1}^K$ until f_θ converges at step M, we hope that $\boldsymbol{\epsilon}$ and $\{\mathbf{G}_k\}_{k=1}^K$ can benefit from the updating capacity of f_θ at every stage.



(a) *text* → *DeCAF*$_6$ (b) *text* → *SIFT*

Figure 3: Classification accuracies (%) on **NUS-WIDE**

Table 1: The data volume for experiments.

	$\lvert\mathbf{X}^{S,c}\rvert$	$\lvert\mathbf{X}^{S,l}\rvert$	$\lvert\mathbf{X}^{S,u}\rvert$	$\lvert\mathbf{X}^{T,c}\rvert$	$\lvert\mathbf{X}^{T,l}\rvert$	$\lvert\mathbf{X}^{T,u}\rvert$	Test
Amazon	2000	2000	9000	2000	100	9000	1900
MRC	500	3000	0	300	30	0	3000
NUS-WIDE	500	5000	0	500	100	0	1000

Table 2: The parameter settings for experiments.

	K	λ	η	α	β	γ	τ
Amazon	3	0.7	1	0.01	10^5	2	0.01
MRC	3	0.7	1	0.01	10^5	2	0.01
NUS-WIDE	3	0.7	1	0.01	10^5	2	10^{-5}

Finally, we can apply $\{\mathbf{z}_i^{T,l}, y_i^{T,l}\}_{i=1}^{n_{T,l}}$ to fine-tune f_θ to improve its performance further, then f_{θ_M} is the classifier we need. We summarize TWLC in Algorithm 1.

Experiments

We compare TLWC with previous HeTL methods on 3 HeTL datasets and conduct a series of experiments to verify its effectiveness on IC data weighting.

Datasets and Parameters

Webis-CLS-10 (Prettenhofer and Stein 2010) is a cross-language sentiment classification dataset, which consists of Amazon product reviews of threes product categories: *book* (B), DVD (D) and *music* (M). The reviews are written in four languages: English (EN), German (GE), French (FR), and Japanese (JP). For each category in a non-English language, 2000 data are translated into English with Google Translate. We adopt the widely used 9 HeTL tasks of this dataset: EFB, EFD, EFM, EGB, EGD, EGM, EJB, EJD, EJM. EFB means, for example, taking *book* reviews in English as the source domain and those in French as the target domain. the documents are represented with TF-IDF and 2000 most frequent words are selected.

Multilingual Reuters Collection (MRC) is a news dataset with five languages (English (EN), French (FR), German (GE), Italian(IT) and Spanish(SP)), where each article is represented by TF-IDF. The news articles in this dataset share 6 topics (C15, CCAT, E21, ECAT, GCAT and M11). Following (Zhou et al. 2016), we take English as the source domain and other languages as target domains, which forms 4 topic classification tasks.

Table 3: Classification accuracies (%) for the 9 cross-language sentiment classification tasks.

Task	TSL	DMMC	HHTL	DFHM	DATN	TLWC$_{sup}$	TLWC
EFB	73.95	76.52	83.63	82.68	78.47	**84.92**	84.50
EFD	74.30	76.23	84.26	83.89	78.63	84.41	**85.00**
EFM	71.15	74.05	83.26	82.26	77.68	83.58	**84.30**
EGB	75.98	77.47	85.42	85.37	77.89	**85.61**	85.57
EGD	76.01	78.28	85.26	85.00	78.16	85.45	**85.57**
EGM	74.57	76.61	84.37	84.47	77.21	84.32	**84.94**
EJB	65.81	68.54	78.26	77.37	71.89	78.05	**79.05**
EJD	70.72	72.12	81.05	81.05	72.42	80.76	**81.68**
EJM	68.22	71.37	79.32	78.26	73.37	79.04	**80.94**

Table 4: Classification accuracies (%) for the 4 cross-language topic classification tasks.

Task	TSL	DMMC	HHTL	DFHM	DATN	TLWC$_{sup}$	TLWC
FR	63.18	65.52	75.93	76.07	70.53	76.47	**78.70**
GE	56.08	58.23	69.47	70.03	65.21	**71.10**	70.08
IT	57.15	60.76	61.80	62.07	55.17	61.77	**68.83**
SP	56.98	62.64	65.80	66.00	63.41	67.37	**69.40**

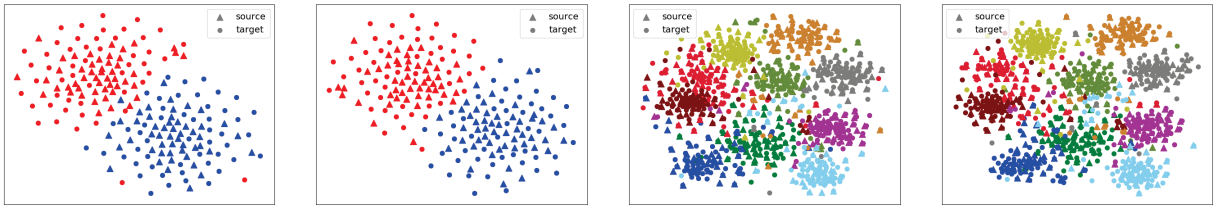
NUS-WIDE (Chua et al. 2009) contains 269,648 images from Flickr and their corresponding text-tag. Follow Wang *et al.* (2018) We conduct experiments on 10 categories: *birds, buildings, car, cat, dog, fish, horses, flowers, mountain and plane*. The source domain data are texts represented by 4096-dimensional tags. The target-domain data are images represented both by 1000-dimensional *DeCAF*$_6$ (Donahue et al. 2014) and 500-dimensional *SIFT* (Lowe 1999) respectively.

The detailed data volume and parameter settings for our experiments are shown in Table 1 and 2.

Baselines

We compare TLWC with the following state-of-the-art HeTL methods based on IC data.

- **TSL** (Xiao and Guo 2013): TSL converts domain bridging into a matrix completion problem. In the matrix, each instance is completed based on IC data. Then a classifer can be employed on the completed matrix.
- **DMMC** (Zhou et al. 2016): DMMC is an extension on TSL. Besides matrix completion, DMMC considers the distribution discrepancy of the two domains.
- **HHTL** (Zhou et al. 2014): HHTL first extracts high-level



(a) HHTL (b) TLWC (c) HHTL (d) DLWC

Figure 4: Feature visualization with t-SNE on EFB and *text*→*DeCAF*$_6$.

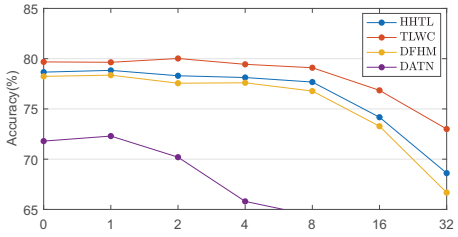
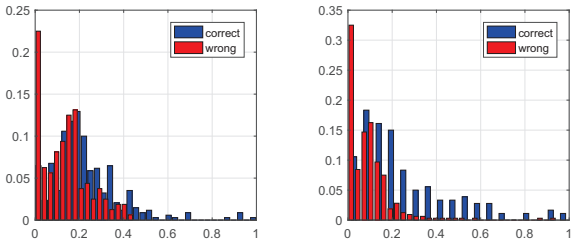


Figure 5: Accuracy vs. n^e



(a) $n^e = 16$ (b) $n^e = 32$

Figure 6: IC data weights distributions.

features for both domains with mSDA and learns feature transformations with IC data features, which projects the features from the target domain to the corresponding layer in the source domain.

- **DHFM** (Guntuku et al. 2016): DHFM also applies high-level-features of IC data to bridge the two domains, while it learns feature transformations in a cross-layer pattern.
- **DATN** (Wang, Cui, and Zhu 2018): DATN establishes two deep autoencoders to learn hidden features for the two domains. Then a feature transformation at top layer is learned to match the two domains.

In addition, to demonstrates the effectiveness of each component, we consider a variant version of TLWC (denoted as TWLC$_{sup}$), which does not update the feature transformation with the meta-learner. The experimental results are shown in Figure 3, Table 3 and Table 4, On **NUS-WIDE**, We do not report the results of TSL and DMMC in Figure 3 because they do not achieve comparable results with other baselines, a possible reasons is that they are designed for cross-language tasks.

We have made significant test in our experiment and the p values for all the tasks are less than 0.01. From the experimental results, we have the following observations: (1) HHTL, DFHM and DATN outperform TSL and DMMC, which demonstrates that high-level features extracted by deep models are more transferable between two domains. (2) HHTL, DFHM perform better than DATN, which shows that the features in middle layers still contain domain-invariant factors. (3) TLWC performs better than all of the state-of-the-art IC data based HeTL methods, which verifies the effectiveness of TLWC on domain adaptation. (4) TLWC$_{sup}$ does not outperforms the baselines on some categories of **NUS-WIDE**, the possible explantation is that compared with machine-translated IC data in **Webis**, the qualities of tag-image IC data of **NUS-WIDE** are much higher, which reduces the necessity to reweight the IC data. In order to verify our guess, we will manually add some noise to these data in the next section. (5) The improvement of TLWC over TLWC$_{sup}$ demonstrates that the meta-optimization over the feature transformation is effective.

Empirical Analysis

Feature Visualization. To validate the transferability of the features learned by TLWC, we randomly select 50 examples form each category of both domains and plot their t-SNE embeddings (van der Maaten and Hinton 2008) learned by HHTL and TLWC in Figure 4. The embedding tasks are EFB and *text* → *DeCAF*$_6$. From Figure 4, we have the following observations: (1) The points with TLWC features are more discriminative than points with HHTL features. (2) With TLWC learned features, the points in one category from the two domains are aligned better.

Noisy Weights Distribution. To statistically prove the effectiveness of TLWC on reweighting IC data, we manually add quantified artificial noise to IC data. In detail, for task *text* → *DeCAF*$_6$, partial IC data are mismatched to other documents and the mismatched number in each category is n^e in average. The performances w.r.t. n^e are shown in Figure 5. To demonstrate the advantage on IC data reweighting of TLWC, we apply TLWC$_{sup}$ to compare with baseline methods.

From Figure 5 we find that the performance gap between the methods is relatively small. However, the performances of previous HeTL methods drop significantly when n^e increases, while the performance of TLWC maintains at a relatively stable level. We plot the weights distribution when

1. It should be a positive tone. **2.** It should be an negative tone.

… étoiles étaient pas mal du tout. … dessin est moins bon, l'humour moins fin. …

… stars were not all bad ① … the design is so good ②, the humor less fine. …

1. Inaccurate phrase. **2.** Personal pronouns cannot be judged.

…Alors je me suis obligée d' aller jusqu'au bout. Il m'a….

……so i forced myself for going through①. He②was….

1. Lack possessive case. **2.** The translation exists grammatical problem. **3.** Phrase meaning does not match the original text.

…roman de Julie Wolkenstein, Colloque sentimental, que certains rapprochaient volontiers de David Lodge, mais là, j'avoue que je n'irai pas jusqu'au bout. …

…previous novel julie wolkenstein ①, colloque sentimental, some willingly approached by david lodge ②, but then, i confess that i will not go through③. …

1. The clause structure is not translated, and the translation exists grammatical problem.

…ne reconnais même plus les personnages.Quant à l'intrigue ..... C'est à se demander si c'est la même personne qui a écrit ce bouquin !!!

…i do not even recognize the characters. as for the plot ..... one wonders if the same person who wrote this book!

1. The translation is accurate.

…Magnifique témoignage à lire. L'histoire est vraiment invraisemblable et tellement dure en même temps…

…Magnificent testimony to read. the story is really unbelievable and so hard at the same time…

1. Some words aren't translated , but overall expression is accurate.

…et une histoire sans queue ni tête mais qui curieusement sent le "déjà vu". …

…a story without beginning or end but curiously feels déjà vu…

1. Some translation exists grammar problem. **2.** The translation simplifies the cumbersome expression into an accurate word.

… les tomes 1 et 3 sont passionnants, mais ce tome n ° 2 est à tel point inutile qu'on peut très bien s'en passer !

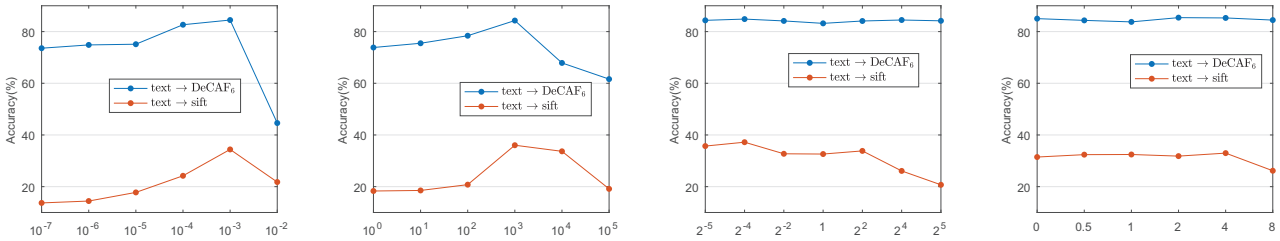
… volumes 1 and 3 are exciting, but volume 2 is so useless that we can very well 's without it!

1. Adjective position is wrong.

…deux personnes ont posté des avis défavorables…, vous trouvez une nuées d'avis hyper positifs et peu argumentés …

……two people have posted negative opinions, …, you see any clouds of opinion and little hyper positive arguments….

Figure 7: IC documents examples of EFB task. 4 pairs with minimum wights are on the left and 4 pairs with maximum weights are on the right.



(a) Accuracy vs. α (b) Accuracy vs. β (c) Accuracy vs. γ (d) Accuracy vs. η

Figure 8: Feature visualization with t-SNE on EFB and *text→DeCAF*$_6$.

$n^e = 16$ and $n^e = 32$ in Figure 6, which illustrates that the weights of the mismatched IC data are mostly pushed to zeros. The experimental results prove that TLWC is able to learn reliable IC data weights according to data qualities, which can advance the domain adaptation of HeTL.

IC Samples Display. To intuitively realize the effectiveness of TLWC on weights learning, we display the 4 pairs of IC documents with minimum weights and 4 pairs with maximum weights of EFB task respectively. The documents are displayed in Figure 7. We invite a volunteer to mark the inappropriate translation in each pair of documents and explained the error above the pair. The volunteer is not told the experimental purpose as well as the data weights. From the figure, we find that there are more translation mistakes in the documents with minimum weights and some mistakes are even on emotions, which is detrimental to the sentiment classification task.

Parameter Sensitivity. We investigate the effects of the parameter α in Eq. 7, β in Eq. 15, γ in Eq. 11 and η in Eq. 9, 14 on tasks based on **NUS-WIDE**. Figure 8 shows the performance variations w.r.t. $\alpha \in \{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$, $\beta \in \{10^0, 10^1, 10^2, 10^3, 10^4, 10^5\}$, $\eta \in \{0, 0.5, 1, 2, 4, 8\}$ and $\gamma \in \{2^{-5}, 2^{-4}, 2^{-2}, 1, 2^2, 2^4, 2^5\}$. the figure shows that the performances w.r.t. α and β exhibit bell-shaped curves. And the choice of $\alpha = 10^5$ and $\beta = 10^4$ would be reasonable in our experiment. Compared with α and β, The accuracies are not so sensitive to the η and γ. For task $text \rightarrow SIFT$, accuracy w.r.t. γ decreases when γ increases. As the weights tend to be zeros or ones when γ is large, this result illustrates that IC data weights with floating values are more robust to learn feature transformations.

Conclusion

In this paper, we proposed a novel Transfer Learning with Weighted Correspondence (TLWC) to perform *heterogeneous* transfer learning with instance-correspondence (IC) data. Different from previous methods that assumed all the IC data are equally important, we construct a meta-learner that utilizes the classification loss in the target domain to guide the IC data weights learning and feature transformation optimization. Based on this framework, the transformed feature space learned by TLWC is more adaptive to the task in the target domain. Extensive experiments on 3 datasets demonstrate the effectiveness of TLWC on IC data weights updating and domain adaptation.

Acknowledgments

This work is supported by National Key R&D Program of China (2018YFC0807500), National Basic Research Program of China (2015CB352300), National Natural Science Foundation of China (No. 61971260, No. 61701273), National Postdoctoral Program for Innovative Talents (No. BX20180172), and the China Postdoctoral Science Foundation (No. 2018M640131).

References

Andrychowicz, M.; Denil, M.; Gomez, S.; Hoffman, M. W.; Pfau, D.; Schaul, T.; Shillingford, B.; and De Freitas, N. 2016. Learning to learn by gradient descent by gradient descent. *neural information processing systems* 3981–3989.

Chen, M.; Xu, Z.; Weinberger, K.; and Sha, F. 2012. Marginalized denoising autoencoders for domain adaptation. *arXiv preprint arXiv:1206.4683*.

Chu, W.; La Torre, F. D.; and Cohn, J. F. 2013. Selective transfer machine for personalized facial action unit detection. 2013:3515–3522.

Chua, T.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. Nus-wide: a real-world web image database from national university of singapore. 48.

Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; and Darrell, T. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

Guntuku, S. C.; Zhou, J. T.; Roy, S.; Lin, W.; and Tsang, I. W. 2016. Understanding deep representations learned in modeling users likes. *IEEE Transactions on Image Processing* 25(8):3762–3774.

Herath, S.; Harandi, M. T.; and Porikli, F. M. 2017. Learning an invariant hilbert space for domain adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 3956–3965.

Hoffman, J.; Rodner, E.; Donahue, J.; Kulis, B.; and Saenko, K. 2014. Asymmetric and category invariant feature transformations for domain adaptation. *International Journal of Computer Vision* 109:28–41.

Jin, X.; He, T.; Wan, C.; Yi, L.; Ding, G.; and Shen, D. 2018. Automatic gating of attributes in deep structure. In *IJCAI*, 2305–2311.

Li, W.; Duan, L.; Xu, D.; and Tsang, I. W. 2013. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36:1134–1148.

Long, M.; Wang, J.; Ding, G.; Sun, J.; and Yu, P. S. 2014. Transfer joint matching for unsupervised domain adaptation. 1410–1417.

Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2017. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning*, 2208–2217.

Lowe, D. G. 1999. Object recognition from local scale-invariant features. In *iccv*, 1150. Ieee.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22(10):1345–1359.

Prettenhofer, P., and Stein, B. 2010. Cross-language text classification using structural correspondence learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, 1118–1127. Association for Computational Linguistics.

Ren, M.; Zeng, W.; Yang, B.; and Urtasun, R. 2018. Learning to reweight examples for robust deep learning. In *ICML*.

Snell, J.; Swersky, K.; and Zemel, R. S. 2017. Prototypical networks for few-shot learning. In *NIPS*.

Tsai, Y.-H.; Yeh, Y.-R.; and Wang, Y.-C. F. 2016. Learning cross-domain landmarks for heterogeneous domain adaptation. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 5081–5090.

Tzeng, E.; Hoffman, J.; Darrell, T.; and Saenko, K. 2015. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, 4068–4076.

van der Maaten, L., and Hinton, G. E. 2008. Visualizing data using t-sne.

Wang, C., and Mahadevan, S. 2011. Heterogeneous domain adaptation using manifold alignment. In *IJCAI 2011*.

Wang, D.; Cui, P.; and Zhu, W. 2018. Deep asymmetric transfer network for unbalanced domain adaptation. In *AAAI*.

Weiss, K.; Khoshgoftaar, T. M.; and Wang, D. 2016. A survey of transfer learning. *Journal of Big Data* 3(1):9.

Xiao, M., and Guo, Y. 2013. A novel two-step method for cross language representation learning. In *NIPS*.

Xiao, M., and Guo, Y. 2015. Feature space independent semi-supervised domain adaptation via kernel matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37:54–66.

Yang, L.; Jing, L.; Yu, J.; and Ng, M. K. 2015. Learning transferred weights from co-occurrence data for heterogeneous transfer learning. *IEEE transactions on neural networks and learning systems* 27(11):2187–2200.

Zhou, J. T.; Pan, S. J.; Tsang, I. W.; and Yan, Y. 2014. Hybrid heterogeneous transfer learning through deep learning. In *AAAI*, 2213–2220.

Zhou, G.; He, T.; Zhao, J.; and Wu, W. 2015. A subspace learning framework for cross-lingual sentiment classification with partial parallel data. In *IJCAI*, 1426–1433.

Zhou, J. T.; Pan, S. J.; Tsang, I. W.; and Ho, S.-S. 2016. Transfer learning for cross-language text categorization through active correspondences construction. In *AAAI*.

Zhu, Y.; Chen, Y.; Lu, Z.; Pan, S. J.; Xue, G.-R.; Yu, Y.; and Yang, Q. 2011. Heterogeneous transfer learning for image classification. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.