Transfer Learning via Learning to Transfer

Ying Wei [1 2] **Yu Zhang** [1] **Junzhou Huang** [2] **Qiang Yang** [1]

Abstract

In transfer learning, what and how to transfer are two primary issues to be addressed, as different transfer learning algorithms applied between a source and a target domain result in different knowledge transferred and thereby the performance improvement in the target domain. Determining the optimal one that maximizes the performance improvement requires either exhaustive exploration or considerable expertise. Meanwhile, it is widely accepted in educational psychology that human beings improve transfer learning skills of deciding what to transfer through meta-cognitive reflection on inductive transfer learning practices. Motivated by this, we propose a novel transfer learning framework known as *Learning to Transfer* (L2T) to automatically determine what and how to transfer are the best by leveraging previous transfer learning experiences. We establish the L2T framework in two stages: 1) we learn a reflection function encrypting transfer learning skills from experiences; and 2) we infer what and how to transfer are the best for a future pair of domains by optimizing the reflection function. We also theoretically analyse the algorithmic stability and generalization bound of L2T, and empirically demonstrate its superiority over several state-of-the-art transfer learning algorithms.

1. Introduction

Inspired by human beings' capabilities to transfer knowledge across tasks, transfer learning aims to leverage knowledge from a source domain to improve the learning performance or minimize the number of labeled examples required in a target domain. It is of particular significance when tackling tasks with limited labeled examples. Transfer learning has proved its wide applicability in, for example, image classification (Long et al., 2015), sentiment classification (Blitzer et al., 2006), dialog systems (Mo et al., 2016), and urban computing (Wei et al., 2016).

Three key research issues in transfer learning, pointed by Pan & Yang, are when to transfer, how to transfer, and what to transfer. Once transfer learning from a source domain is considered to benefit a target domain (when to transfer), an algorithm (how to transfer) discovers the transferable knowledge across domains (what to transfer). Different algorithms are likely to discover different transferable knowledge, and thereby lead to uneven transfer learning effectiveness which is evaluated by the performance improvement over non-transfer baselines in a target domain. To achieve the optimal performance improvement for a target domain given a source domain, researchers may try tens to hundreds of transfer learning algorithms covering instance (Dai et al., 2007), parameter (Tommasi et al., 2014), and feature (Pan et al., 2011) based algorithms. Such brute-force exploration is computationally expensive and practically impossible. As a tradeoff, a sub-optimal improvement is usually obtained from a heuristically selected algorithm, which unfortunately requires considerable expertise in an ad-hoc and unsystematic manner.

Exploring different algorithms is not the only way to optimize what to transfer. Previous transfer learning experiences do also help, which has been widely accepted in educational psychology (Luria, 1976; Belmont et al., 1982). Human beings sharpen transfer learning skills of deciding what to transfer by conducting meta-cognitive reflection on diverse transfer learning experiences. For example, children who are good at playing chess may transfer mathematical skills, visuospatial skills, and decision making skills learned from chess to solve arithmetic problems, to solve pattern matching puzzles, and to play basketball, respectively. At a later age, it will be easier for them to decide to transfer mathematical and decision making skills learned from chess, rather than visuospatial skills, to market investment. Unfortunately, all existing transfer learning algorithms transfer from scratch and ignore previous transfer learning experiences.

Motivated by this, we propose a novel transfer learning framework called Learning to Transfer (L2T). The key idea of the L2T is to enhance the transfer learning effectiveness from a source to a target domain by leveraging previous

[1] Hong Kong University of Science and Technology, Hong Kong [2] Tencent AI Lab, Shenzhen, China. Correspondence to: Ying Wei <judyweiying@gmail.com>, Qiang Yang <qyang@cse.ust.hk>.

Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

transfer learning experiences to optimize what and how to transfer between them. To achieve the goal, we establish the L2T in two stages. During the first stage, we encode each transfer learning experience into three components: a pair of source and target domains, the transferred knowledge between them parameterized as latent feature factors, and performance improvement. We learn from all experiences a reflection function which maps a pair of domains and the transferred knowledge between them to the performance improvement. The reflection function, therefore, is believed to encrypt transfer learning skills of deciding what and how to transfer. In the second stage, what to transfer between a newly arrived pair of domains is optimized so that the value of the learned reflection function, matching to the performance improvement, is maximized.

The contribution of this paper lies in that we propose a novel transfer learning framework which opens a new door to improve transfer learning effectiveness by taking advantage of previous transfer learning experiences. The L2T can discover more transferable knowledge in a systematic and automatic fashion without requiring considerable expertise. We have also provided theoretic analyses to its algorithmic stability and generalization bound, and conducted comprehensive empirical studies showing the L2T's superiority over state-of-the-art transfer learning algorithms.

2. Related Work

Transfer Learning Pan & Yang identified three key research issues in transfer learning as what, how, and when to transfer. Parameters (Yang et al., 2007a; Tommasi et al., 2014), instances (Dai et al., 2007), or latent feature factors (Pan et al., 2011) can be transferred between domains. A few works (Yang et al., 2007a; Tommasi et al., 2014) transfer *parameters* from source domains to regularize parameters of SVM-based models in a target domain. In (Dai et al., 2007), a basic learner in a target domain is boosted by borrowing the most useful source *instances*. Various techniques capable of learning transferable *latent feature factors* between domains have been investigated extensively. These techniques include manually selected pivot features (Blitzer et al., 2006), dimension reduction (Pan et al., 2011; Baktashmotlagh et al., 2013; 2014), collective matrix factorization (Long et al., 2014), dictionary learning and sparse coding (Raina et al., 2007; Zhang et al., 2016), manifold learning (Gopalan et al., 2011; Gong et al., 2012), and deep learning (Yosinski et al., 2014; Long et al., 2015; Tzeng et al., 2015). Unlike L2T, all existing transfer learning studies transfer from scratch, i.e., only considering the pair of domains of interest but ignoring previous transfer learning experiences. Better yet, L2T can even collect all algorithms' wisdom together, considering that any algorithm mentioned above can be applied in a transfer learning experience.

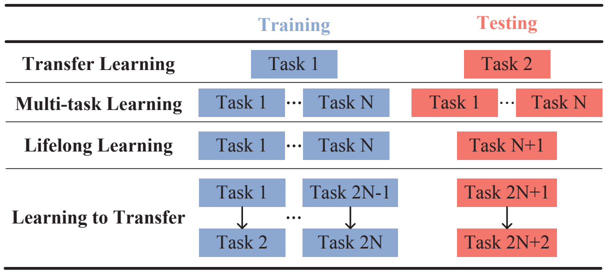


Figure 1. Illustration of the differences between our work and the other three lines of work.

Multi-task Learning Multi-task learning (Caruana, 1997; Argyriou et al., 2007) trains multiple related tasks simultaneously and learns shared knowledge among tasks, so that all tasks reinforce each other in generalization abilities. However, multi-task learning assumes that training and testing examples follow the same distribution, as Figure 1 shows, which is different from transfer learning we focus on.

Lifelong Learning Assuming a new learning task to lie in the same environment as training tasks, learning to learn (Thrun & Pratt, 1998) or meta-learning (Maurer, 2005; Finn et al., 2017) transfers the knowledge shared among training tasks to the new task. (Ruvolo & Eaton, 2013; Pentina & Lampert, 2015) consider lifelong learning as online meta-learning. Though L2T and lifelong (meta) learning both aim to improve a learning system by leveraging histories, L2T differs from them in that each historical experience we consider is a transfer learning task rather than a traditional learning task as Figure 1 illustrates. Thus we learn transfer learning skills instead of task-sharing knowledge.

3. Learning to Transfer

We begin by first briefing the proposed L2T framework. Then we detail the two stages in L2T, i.e., learning transfer learning skills from previous transfer learning experiences and applying those skills to infer what and how to transfer for a future pair of source and target domains.

3.1. The L2T Framework

A L2T agent previously conducted transfer learning several times, and kept a record of N_e transfer learning experiences. We define each transfer learning experience as $E_e = (\langle \mathcal{S}_e, \mathcal{T}_e \rangle, a_e, l_e)$ in which $\mathcal{S}_e = \{\mathbf{X}_e^s, \mathbf{y}_e^s\}$ and $\mathcal{T}_e = \{\mathbf{X}_e^t, \mathbf{y}_e^t\}$ denote a source domain and a target domain, respectively. $\mathbf{X}_e^* \in \mathbb{R}^{n_e^* \times m}$ represents the feature matrix if either domain has n_e^* examples in a m-dimensional feature space $\mathcal{X}_e^*$, where the superscript $*$ can be either s or t to denote a source or a target domain. $\mathbf{y}_e^* \in \mathcal{Y}_e^*$ denotes the vector of labels with the length being n_{le}^*. The number of target labeled examples is much smaller than that of source labeled examples, i.e., $n_{le}^t \ll n_{le}^s$. We focus on the setting $\mathcal{X}_e^s = \mathcal{X}_e^t$ and $\mathcal{Y}_e^s \neq \mathcal{Y}_e^t$ for each pair of domains.

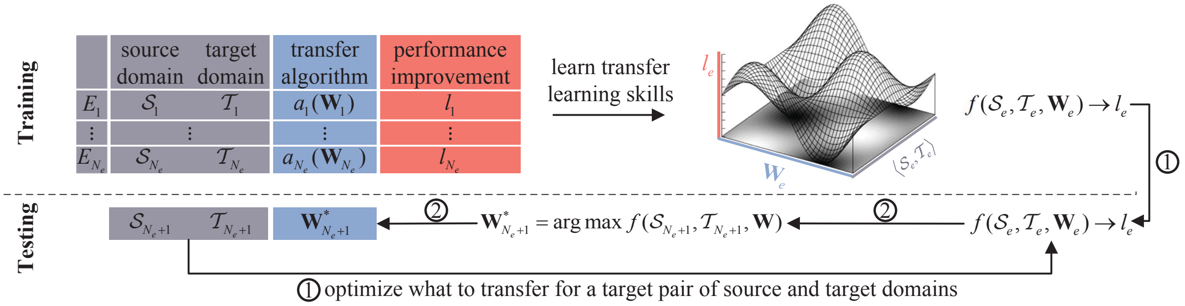

Training
source domain
target domain
transfer algorithm
performance improvement
E_1 $\mathcal{S}_1$ $\mathcal{T}_1$ $a_1(\mathbf{W}_1)$ l_1
⋮
E_{N_e} $\mathcal{S}_{N_e}$ $\mathcal{T}_{N_e}$ $a_{N_e}(\mathbf{W}_{N_e})$ l_{N_e}
learn transfer learning skills
l_e
$\mathbf{W}_e$
$\langle\mathcal{S}_e,\mathcal{T}_e\rangle$
$f(\mathcal{S}_e,\mathcal{T}_e,\mathbf{W}_e)\rightarrow l_e$
①
Testing
$\mathcal{S}_{N_e+1}$ $\mathcal{T}_{N_e+1}$
$\mathbf{W}^*_{N_e+1}$
②
$\mathbf{W}^*_{N_e+1}=\arg\max f(\mathcal{S}_{N_e+1},\mathcal{T}_{N_e+1},\mathbf{W})$
②
$f(\mathcal{S}_e,\mathcal{T}_e,\mathbf{W}_e)\rightarrow l_e$
① optimize what to transfer for a target pair of source and target domains

Figure 2. Illustration of the L2T framework: in the training stage, we have N_e transfer learning experiences $\{E_1,\cdots,E_{N_e}\}$ from which we learn a reflection function f encrypting transfer learning skills; in the testing stage, given the (N_e+1)-th source-target pair and the learned reflection function f(①), we optimize the transferred knowledge between them, i.e., $\mathbf{W}^*_{N_e+1}$, by maximizing the value of f(②).

$a_e\in\mathcal{A}=\{a_1,\cdots,a_{N_a}\}$ denotes a transfer learning algorithm having been applied between $\mathcal{S}_e$ and $\mathcal{T}_e$. Suppose that the transferred knowledge by the algorithm a_e can be parameterized as $\mathbf{W}_e$. Finally, each transfer learning experience is labeled by the performance improvement ratio $l_e=p_e^{st}/p_e^t$, where p_e^t is the learning performance (e.g., classification accuracy) on a test dataset in $\mathcal{T}_e$ without transfer and p_e^{st} is that on the same test dataset after transferring $\mathbf{W}_e$ from $\mathcal{S}_e$.

With N_e transfer learning experiences $\{E_1,\cdots,E_{N_e}\}$ as the input, the L2T agent learns a function f such that $f(\mathcal{S}_e,\mathcal{T}_e,\mathbf{W}_e)$ approximates l_e as shown in the training stage of Figure 2. We call f a *reflection* function which encrypts meta-cognitive transfer learning skills - what and how to transfer can maximize the improvement ratio given a pair of domains. Whenever a new pair of domains $\langle\mathcal{S}_{N_e+1},\mathcal{T}_{N_e+1}\rangle$ arrives, the L2T agent can optimize the knowledge to be transferred, i.e., $\mathbf{W}^*_{N_e+1}$, by maximizing the value of f (see step ② of the testing stage in Figure 2).

3.2. Parameterizing What to Transfer

Transfer learning algorithms applied can vary from experience to experience. Uniformly parameterizing "what to transfer" for any algorithm out of the base algorithm set $\mathcal{A}$ is a prerequisite for learning the reflection function. In this work, we consider $\mathcal{A}$ to contain algorithms transferring single-level latent feature factors, because existing parameter-based and instance-based algorithms cannot address the transfer learning setting we focus on (i.e., $\mathcal{X}_s^e=\mathcal{X}_t^e$ and $\mathcal{Y}_s^e\neq\mathcal{Y}_t^e$). Though limited parameter-based algorithms (Yang et al., 2007a; Tommasi et al., 2014) can transfer across domains in heterogeneous label spaces, they can only handle binary classification problems. Deep neural network based algorithms (Yosinski et al., 2014; Long et al., 2015; Tzeng et al., 2015) transferring latent feature factors in multiple levels are left for our future research. As a result, we parameterize what to transfer with a latent feature factor matrix $\mathbf{W}$ which is elaborated in the following.

Latent feature factor based algorithms aim to learn domain-invariant feature factors across domains. Consider classifying dog pictures as a source domain and cat pictures as a target domain. The domain-invariant feature factors may include eyes, mouth, tails, etc. What to transfer, in this case, is the shared feature factors across domains. The way of defining domain-invariant feature factors dictates two groups of latent feature factor based algorithms, i.e., common latent space based and manifold ensemble based algorithms.

Common Latent Space Based This line of algorithms, including but not limited to TCA (Pan et al., 2011), LSDT (Zhang et al., 2016), and DIP (Baktashmotlagh et al., 2013), assumes that domain-invariant feature factors lie in a single shared latent space. We denote by φ the function mapping original feature representation into the latent space. If φ is linear, it can be represented as an embedding matrix $\mathbf{W}\in\mathbb{R}^{m\times u}$ where u is the dimensionality of the latent space. Therefore, we can parameterize what to transfer we focus on with $\mathbf{W}$ which describes u latent feature factors. Otherwise, if φ is nonlinear, what to transfer can still be parameterized with $\mathbf{W}$. Though a nonlinear φ is not explicitly specified in most cases such as LSDT using sparse coding, target examples represented in the latent space $\mathbf{Z}_e^t=\varphi(\mathbf{X}_e^t)\in\mathbb{R}^{n_e^t\times u}$ are always available. Consequently, we obtain the similarity metric matrix (Cao et al., 2013) in the latent space, i.e., $\mathbf{G}=(\mathbf{X}_e^t)^\dagger\mathbf{Z}_e^t(\mathbf{Z}_e^t)^T[(\mathbf{X}_e^t)^T]^\dagger\in\mathbb{R}^{m\times m}$ according to $\mathbf{X}_e^t\mathbf{G}(\mathbf{X}_e^t)^T=\mathbf{Z}_e^t(\mathbf{Z}_e^t)^T$, where $(\mathbf{X}_e^t)^\dagger$ is the pseudo-inverse of $\mathbf{X}_e^t$. LDL decomposition on $\mathbf{G}=\mathbf{LDL}^T$ brings the latent feature factor matrix $\mathbf{W}=\mathbf{LD}^{1/2}$.

Manifold Ensemble Based Initiated by Gopalan et al., manifold ensemble based algorithms consider that a source and a target domain share multiple subspaces (of the same dimension) as points on the Grassmann manifold between them. The representation of target examples on u domain-invariant latent factors turns to $\mathbf{Z}_e^{t(n_u)}=[\varphi_1(\mathbf{X}_e^t),\cdots,\varphi_{n_u}(\mathbf{X}_e^t)]\in\mathbb{R}^{n_e^t\times n_u u}$, if n_u subspaces on the manifold are sampled. When all continuous subspaces on the manifold are sampled, i.e., $n_u\rightarrow\infty$, Gong et al. proved that $\mathbf{Z}_e^{t(\infty)}(\mathbf{Z}_e^{t(\infty)})^T = \mathbf{X}_e^t \mathbf{G} (\mathbf{X}_e^t)^T$ where $\mathbf{G}$ is the similarity metric matrix. For computational details of $\mathbf{G}$, please refer to (Gong et al., 2012). $\mathbf{W} = \mathbf{L}\mathbf{D}^{1/2}$ with $\mathbf{L}$ and $\mathbf{D}$ obtained from performing LDL decomposition on $\mathbf{G} = \mathbf{L}\mathbf{D}\mathbf{L}^T$, therefore, is also qualified to represent latent feature factors distributed in a series of subspaces on a manifold.

3.3. Learning from Experiences

The goal here is to learn a reflection function f such that $f(\mathcal{S}_e, \mathcal{T}_e, \mathbf{W}_e)$ can approximate l_e for all experiences $\{E_1, \cdots, E_{N_e}\}$. The improvement ratio l_e is closely related to two aspects: 1) the difference between a source and a target domain in the shared latent space, and 2) the discriminative ability of a target domain in the latent space. The smaller difference guarantees more overlap between domains in the latent space, which signifies more transferable latent feature factors and higher improvement ratios as a result. The discriminative ability of a target domain in the latent space is also vital to improve performances. Therefore, we build f to take both aspects into consideration.

The Difference between a Source and a Target Domain We follow (Pan et al., 2011) and adopt the maximum mean discrepancy (MMD) (Gretton et al., 2012b) to measure the difference between domains. By mapping two domains into the reproducing kernel Hilbert space (RKHS), MMD empirically evaluates the distance between the mean of source examples and that of target examples:

$$\begin{aligned}
&\hat{d}_e^2(\mathbf{X}_e^s\mathbf{W}_e, \mathbf{X}_e^t\mathbf{W}_e)\\
=&\left\| \frac{1}{n_e^s}\sum_{i=1}^{n_e^s}\phi(\mathbf{x}_{ei}^s\mathbf{W}_e) - \frac{1}{n_e^t}\sum_{j=1}^{n_e^t}\phi(\mathbf{x}_{ej}^t\mathbf{W}_e)\right\|_{\mathcal{H}}^2\\
=&\frac{1}{(n_e^s)^2}\sum_{i,i'=1}^{n_e^s}\mathcal{K}(\mathbf{x}_{ei}^s\mathbf{W}_e, \mathbf{x}_{ei'}^s\mathbf{W}_e)\\
+&\frac{1}{(n_e^t)^2}\sum_{j,j'=1}^{n_e^t}\mathcal{K}(\mathbf{x}_{ej}^t\mathbf{W}_e, \mathbf{x}_{ej'}^t\mathbf{W}_e)\\
-&\frac{2}{n_e^s n_e^t}\sum_{i,j=1}^{n_e^s,n_e^t}\mathcal{K}(\mathbf{x}_{ei}^s\mathbf{W}_e, \mathbf{x}_{ej}^t\mathbf{W}_e), \qquad (1)
\end{aligned}$$

where $\mathbf{x}_{ej}^t$ is the j-th example in $\mathbf{X}_e^t$, and ϕ maps from the u-dimensional latent space to the RKHS $\mathcal{H}$. $\mathcal{K}(\cdot,\cdot) = \langle\phi(\cdot),\phi(\cdot)\rangle$ is the kernel function. Different kernels $\mathcal{K}$ lead to different MMD distances and thereby different values of f. Thus learning the reflection function f is equivalent to optimizing $\mathcal{K}$ so that the MMD distance can well characterize the improvement ratio l_e for all pairs of domains. Inspired by multi-kernel MMD (Gretton et al., 2012b), we parameterize $\mathcal{K}$ as a linear combination of N_k PSD kernels, i.e., $\mathcal{K} = \sum_{k=1}^{N_k}\beta_k\mathcal{K}_k$ ($\beta_k \geq 0, \forall k$), and learn the coefficients $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_{N_k}]$ instead. Using $\boldsymbol{\beta}$, the MMD can be rewritten as $\hat{d}_e^2(\mathbf{X}_e^s\mathbf{W}_e, \mathbf{X}_e^t\mathbf{W}_e) = \sum_{k=1}^{N_k}\beta_k\hat{d}_{e(k)}^2(\mathbf{X}_e^s\mathbf{W}_e, \mathbf{X}_e^t\mathbf{W}_e) = \boldsymbol{\beta}^T\hat{\mathbf{d}}_e$, where $\hat{\mathbf{d}}_e = [\hat{d}_{e(1)}^2, \cdots, \hat{d}_{e(N_k)}^2]$ with $\hat{d}_{e(k)}^2$ computed by the k-th kernel $\mathcal{K}_k$. In this paper, we consider RBF kernels $\mathcal{K}_k(\mathbf{a},\mathbf{b}) = \exp(-\|\mathbf{a}-\mathbf{b}\|^2/\delta_k)$ by varying the bandwidth δ_k.

Unfortunately, the MMD alone is insufficient to measure the difference between domains. The distance variance among all pairs of instances across domains is also required to fully characterize the difference. A pair of domains with small MMD but extremely high variance still have little overlap. Equation (1) is actually the empirical estimation of $d_e^2(\mathbf{X}_e^s\mathbf{W}_e, \mathbf{X}_e^t\mathbf{W}_e) = \mathbf{E}_{\mathbf{x}_e^s\mathbf{x}_e^{s\prime}\mathbf{x}_e^t\mathbf{x}_e^{t\prime}} h(\mathbf{x}_e^s, \mathbf{x}_e^{s\prime}, \mathbf{x}_e^t, \mathbf{x}_e^{t\prime})$ (Gretton et al., 2012b) where $h(\mathbf{x}_e^s, \mathbf{x}_e^{s\prime}, \mathbf{x}_e^t, \mathbf{x}_e^{t\prime}) = \mathcal{K}(\mathbf{x}_e^s\mathbf{W}_e, \mathbf{x}_e^{s\prime}\mathbf{W}_e) + \mathcal{K}(\mathbf{x}_e^t\mathbf{W}_e, \mathbf{x}_e^{t\prime}\mathbf{W}_e) - \mathcal{K}(\mathbf{x}_e^s\mathbf{W}_e, \mathbf{x}_e^{t\prime}\mathbf{W}_e) - \mathcal{K}(\mathbf{x}_e^{s\prime}\mathbf{W}_e, \mathbf{x}_e^t\mathbf{W}_e)$. Consequently, the distance variance, σ_e^2, equals

$$\begin{aligned}
\sigma_e^2(\mathbf{X}_e^s\mathbf{W}_e, \mathbf{X}_e^t\mathbf{W}_e) = &\mathbf{E}_{\mathbf{x}_e^s\mathbf{x}_e^{s\prime}\mathbf{x}_e^t\mathbf{x}_e^{t\prime}}[(h(\mathbf{x}_e^s, \mathbf{x}_e^{s\prime}, \mathbf{x}_e^t, \mathbf{x}_e^{t\prime})\\
&- \mathbf{E}_{\mathbf{x}_e^s\mathbf{x}_e^{s\prime}\mathbf{x}_e^t\mathbf{x}_e^{t\prime}} h(\mathbf{x}_e^s, \mathbf{x}_e^{s\prime}, \mathbf{x}_e^t, \mathbf{x}_e^{t\prime}))^2].
\end{aligned}$$

To be consistent with the MMD characterized with N_k PSD kernels, we rewrite $\sigma_e^2 = \boldsymbol{\beta}^T\mathbf{Q}_e\boldsymbol{\beta}$ where $\mathbf{Q}_e = \text{cov}(h) = \begin{bmatrix} \sigma_{e(1,1)} & \cdots & \sigma_{e(1,N_k)} \\ \cdots & \cdots & \cdots \\ \sigma_{e(N_k,1)} & \cdots & \sigma_{e(N_k,N_k)} \end{bmatrix}$. Each element $\sigma_{e(k_1,k_2)} = \text{cov}(h_{k_1}, h_{k_2}) = \mathbf{E}[(h_{k_1} - \mathbf{E}h_{k_1})(h_{k_2} - \mathbf{E}h_{k_2})]$. Note that $\mathbf{E}h_{k_1}$ is shorthand for $\mathbf{E}_{\mathbf{x}_e^s\mathbf{x}_e^{s\prime}\mathbf{x}_e^t\mathbf{x}_e^{t\prime}} h_{k1}(\mathbf{x}_e^s, \mathbf{x}_e^{s\prime}, \mathbf{x}_e^t, \mathbf{x}_e^{t\prime})$ where h_{k_1} is calculated using the k_1-th kernel. We detail the empirical estimate $\hat{\mathbf{Q}}_e$ of $\mathbf{Q}_e$ in the supplementary due to page limit.

The Discriminative Ability of a Target Domain In view of limited labeled examples in a target domain, we resort to unlabeled examples to evaluate the discriminative ability. The principles of the unlabeled discriminant criterion are two-fold: 1) similar examples should still be neighbours after being embedded into the latent space; and 2) dissimilar examples should be far away. We adopt the unlabeled discriminant criterion proposed in (Yang et al., 2007b),

$$\tau_e = \text{tr}(\mathbf{W}_e^T\mathbf{S}_e^N\mathbf{W}_e)/\text{tr}(\mathbf{W}_e^T\mathbf{S}_e^L\mathbf{W}_e),$$

where $\mathbf{S}_e^L = \sum_{j,j'=1}^{n_e^t}\frac{H_{jj'}}{(n_e^t)^2}(\mathbf{x}_{ej}^t - \mathbf{x}_{ej'}^t)(\mathbf{x}_{ej}^t - \mathbf{x}_{ej'}^t)^T$ is the local scatter covariance matrix with the neighbour information $H_{jj'}$ defined as $H_{jj'} = \begin{cases} \mathcal{K}(\mathbf{x}_{ej}^t, \mathbf{x}_{ej'}^t), & \text{if} \quad \mathbf{x}_{ej}^t \in \mathcal{N}_r(\mathbf{x}_{ej'}^t) \quad \text{and} \quad \mathbf{x}_{ej'}^t \in \mathcal{N}_r(\mathbf{x}_{ej}^t) \\ 0, & \text{otherwise} \end{cases}$.
If $\mathbf{x}_{ej}^t$ and $\mathbf{x}_{ej'}^t$ are mutual r-nearest neighbours to each other, $H_{jj'}$ equals the kernel value $\mathcal{K}(\mathbf{x}_{ej}^t, \mathbf{x}_{ej'}^t)$. By maximizing the unlabeled discriminant criterion τ_e, the local scatter covariance matrix guarantees the first principle, while $\mathbf{S}_e^N = \sum_{j,j'=1}^{n_e^t}\frac{\mathcal{K}(\mathbf{x}_{ej}^t, \mathbf{x}_{ej'}^t) - H_{jj'}}{(n_e^t)^2}(\mathbf{x}_{ej}^t - \mathbf{x}_{ej'}^t)(\mathbf{x}_{ej}^t - \mathbf{x}_{ej'}^t)^T$, the non-local scatter covariance matrix, enforces the second principle. τ_e also depends on kernels which in this case indicate different neighbour information and different degrees of similarity between neighboured examples. With $\tau_{e(k)}$ obtained from the k-th kernel $\mathcal{K}_k$, the unlabeled discriminant criterion τ_e can be written as $\tau_e = \sum_{k=1}^{N_k}\beta_k\tau_{e(k)} = \boldsymbol{\beta}^T\boldsymbol{\tau}_e$ where $\boldsymbol{\tau}_e = [\tau_{e(1)}, \cdots, \tau_{e(N_k)}]$.

The Optimization Problem Combining the two aspects abovementioned to model the reflection function f, we finally formulate the optimization problem as follows,

$$\begin{aligned}&\boldsymbol{\beta}^*, \lambda^*, \mu^*, b^* = \\ &\arg\min_{\boldsymbol{\beta},\lambda,\mu,b} \sum_{e=1}^{N_e} \mathcal{L}_h\left(\boldsymbol{\beta}^T \hat{\mathbf{d}}_e + \lambda\boldsymbol{\beta}^T\hat{\mathbf{Q}}_e\boldsymbol{\beta} + \frac{\mu}{\boldsymbol{\beta}^T\boldsymbol{\tau}_e} + b, \frac{1}{l_e}\right) \\ &\qquad + \gamma_1 R(\boldsymbol{\beta}, \lambda, \mu, b), \\ &\text{s.t.} \quad \beta_k \geq 0,\ \forall k \in \{1, \cdots, N_k\},\ \lambda \geq 0,\ \mu \geq 0, \end{aligned} \tag{2}$$

where $1/f = \boldsymbol{\beta}^T \hat{\mathbf{d}}_e + \lambda\boldsymbol{\beta}^T\hat{\mathbf{Q}}_e\boldsymbol{\beta} + \frac{\mu}{\boldsymbol{\beta}^T\boldsymbol{\tau}_e} + b$ and $\mathcal{L}_h(\cdot)$ is the Huber regression loss (Huber et al., 1964) constraining the value of $1/f$ to be as close to $1/l_e$ as possible. γ_1 controls the complexity of the parameters by l2-regularization. Minimizing the difference between domains, including the MMD distance $\boldsymbol{\beta}^T\hat{\mathbf{d}}_e$ and the distance variance $\boldsymbol{\beta}^T\hat{\mathbf{Q}}_e\boldsymbol{\beta}$, and meanwhile maximizing the discriminant criterion $\boldsymbol{\beta}^T\boldsymbol{\tau}_e$ in the target domain will contribute a large performance improvement ratio l_e (i.e., a small $1/l_e$). λ and μ balance the importance of the three terms in f, and b is the bias term.

3.4. Inferring What to Transfer

Once the L2T agent has learned the reflection function $f(\mathcal{S}, \mathcal{T}, \mathbf{W}; \boldsymbol{\beta}^*, \lambda^*, \mu^*, b^*)$, it takes advantage of the function to optimize what to transfer, i.e., the latent feature factor matrix $\mathbf{W}$, for a newly arrived source domain $\mathcal{S}_{N_e+1}$ and a target domain $\mathcal{T}_{N_e+1}$. The optimal latent feature factor matrix $\mathbf{W}^*_{N_e+1}$ should maximize the value of f. To this end, we optimize the following objective with regard to $\mathbf{W}$,

$$\begin{aligned}\mathbf{W}^*_{N_e+1} &= \arg\max_{\mathbf{W}}\ f(\mathcal{S}_{N_e+1}, \mathcal{T}_{N_e+1}, \mathbf{W}; \boldsymbol{\beta}^*, \lambda^*, \mu^*, b^*) - \gamma_2\|\mathbf{W}\|_F^2 \\ &= \arg\min_{\mathbf{W}} (\boldsymbol{\beta}^*)^T\hat{\mathbf{d}}_{\mathbf{W}} + \lambda^*(\boldsymbol{\beta}^*)^T\hat{\mathbf{Q}}_{\mathbf{W}}\boldsymbol{\beta}^* + \mu^*\frac{1}{(\boldsymbol{\beta}^*)^T\boldsymbol{\tau}_{\mathbf{W}}} \\ &\qquad + \gamma_2\|\mathbf{W}\|_F^2, \end{aligned} \tag{3}$$

where $\|\cdot\|_F$ denotes the matrix Frobenius norm and γ_2 controls the complexity of $\mathbf{W}$. The first and second terms in problem (3) can be calculated as

$$\begin{aligned}(\boldsymbol{\beta}^*)^T\hat{\mathbf{d}}_{\mathbf{W}} &= \sum_{k=1}^{N_k}\beta_k^*\Bigg[\frac{1}{a^2}\sum_{i,i'=1}^{a}\mathcal{K}_k(\mathbf{v}_i\mathbf{W}, \mathbf{v}_{i'}\mathbf{W}) + \\ &\frac{1}{b^2}\sum_{j,j'=1}^{b}\mathcal{K}_k(\mathbf{w}_j\mathbf{W}, \mathbf{w}_{j'}\mathbf{W}) - \frac{2}{ab}\sum_{i,j=1}^{a,b}\mathcal{K}_k(\mathbf{v}_i\mathbf{W}, \mathbf{w}_j\mathbf{W})\Bigg],\end{aligned}$$

$$\begin{aligned}&(\boldsymbol{\beta}^*)^T\hat{\mathbf{Q}}_{\mathbf{W}}\boldsymbol{\beta}^* = \frac{1}{n^2-1}\sum_{i,i'=1}^{n}\sum_{k=1}^{N_k}\Bigg\{\beta_k^*\Bigg[\mathcal{K}_k(\mathbf{v}_i\mathbf{W}, \mathbf{v}_{i'}\mathbf{W}) + \\ &\mathcal{K}_k(\mathbf{w}_i\mathbf{W}, \mathbf{w}_{i'}\mathbf{W}) - 2\mathcal{K}_k(\mathbf{v}_i\mathbf{W}, \mathbf{w}_{i'}\mathbf{W}) - \frac{1}{n^2}\sum_{i,i'=1}^{n}\Bigg(\mathcal{K}_k(\mathbf{v}_i\mathbf{W}, \mathbf{v}_{i'}\mathbf{W}) \\ &+ \mathcal{K}_k(\mathbf{w}_i\mathbf{W}, \mathbf{w}_{i'}\mathbf{W}) - 2\mathcal{K}_k(\mathbf{v}_i\mathbf{W}, \mathbf{w}_{i'}\mathbf{W})\Bigg)\Bigg]\Bigg\}^2,\end{aligned}$$

where the shorthand $\mathbf{v}_i = \mathbf{x}^s_{(N_e+1)i}$, $\mathbf{v}_{i'} = \mathbf{x}^s_{(N_e+1)i'}$, $\mathbf{w}_j = \mathbf{x}^t_{(N_e+1)j}$, $\mathbf{w}_{j'} = \mathbf{x}^t_{(N_e+1)j'}$, $a = n^s_{N_e+1}$, and $b = n^t_{N_e+1}$ are used due to space limit. Note that $n = \min(n^s_{N_e+1}, n^t_{N_e+1})$. The third term in problem (3) can be computed as $(\boldsymbol{\beta}^*)^T\boldsymbol{\tau}_{\mathbf{W}} = \sum_{k=1}^{N_k}\beta_k^*\frac{\mathrm{tr}(\mathbf{W}^T\mathbf{S}_k^N\mathbf{W})}{\mathrm{tr}(\mathbf{W}^T\mathbf{S}_k^L\mathbf{W})}$. We optimize the non-convex problem (3) w.r.t $\mathbf{W}$ by employing a conjugate gradient method in which the gradient is listed in the supplementary material.

4. Stability and Generalization Bounds

In this section, we would theoretically investigate how previous transfer learning experiences influence a transfer learning task of interest. We also provide and prove the algorithmic stability and generalization bound for latent feature factor based transfer learning algorithms without experiences considered in the supplementary.

Consider $\mathbf{S} = \{\langle\mathcal{S}_1, \mathcal{T}_1\rangle, \cdots, \langle\mathcal{S}_{N_e}, \mathcal{T}_{N_e}\rangle\}$ to be N_e transfer learning experiences or the so-called meta-samples (Maurer, 2005). Let $\mathbf{L}(\mathbf{S})$ be our algorithm that learns meta-cognitive knowledge from N_e transfer learning experiences in $\mathbf{S}$ and applies the knowledge to the (N_e+1)-th transfer learning task $\langle\mathcal{S}_{N_e+1}, \mathcal{T}_{N_e+1}\rangle$. To analyse the stability and give the generalization bound, we make an assumption on the distribution from which all N_e transfer learning experiences as meta-samples are sampled. For every environment $\mathcal{E}$ we have, all N_e pairs of source and target domains in $\mathbf{S}$ are drawn according to an algebraic β-mixing stationary distribution $(\mathbf{D}_{\mathcal{E}})^{N_e}$, which is not i.i.d.. Intuitively, the algebraical β-mixing stationary distribution (see Definition 2 in (Mohri & Rostamizadeh, 2010)) with the β-mixing coefficient $\beta(m) \leq \beta_0/m^r$ models the dependence between future samples and past samples by a distance of at least m. The independent block technique (Bernstein, 1927) has been widely adopted to deal with non-i.i.d. learning problems. Under this assumption, $\mathbf{L}(\mathbf{S})$ is uniformly stable.

Theorem 1. *Suppose that for any $\mathbf{x}^t_e$ and for any y^t_e we have $\|\mathbf{x}^t_e\|^2 \leq r_x$ and $|y^t_e| \leq B$. Meanwhile, for any e-th transfer learning experience, we assume that the latent feature factor matrix $\|\mathbf{W}_e\| \leq r_W$. To meet the assumption above, we reasonably simplify $\mathbf{L}(\mathbf{S})$ so that the latent feature factor matrix for the (N_e+1)-th transfer learning task is a linear combination of all N_e historical latent factor feature matrices plus a noisy latent feature matrix $\mathbf{W}_\epsilon$ satisfying $\|\mathbf{W}_\epsilon\| \leq r_\epsilon$, i.e., $\mathbf{W}_{N_e+1} = \sum_{e=1}^{N_e} c_e\mathbf{W}_e + \mathbf{W}_\epsilon$ with each coefficient $0 \leq c_e \leq 1$. Our algorithm $\mathbf{L}(\mathbf{S})$ is uniformly stable. For any $\langle\mathcal{S}, \mathcal{T}\rangle$ as the coming transfer learning task, the following inequality holds:*

$$\begin{aligned}&\left|l_{emp}(\mathbf{L}(\mathbf{S}), (\mathcal{S}, \mathcal{T})) - l_{emp}(\mathbf{L}(\mathbf{S}^{e_0}), (\mathcal{S}, \mathcal{T}))\right| \\ &\leq \frac{4(4N_e - 3 + r_\epsilon/r_W)B^2r_x}{\lambda N_e^2} \sim \mathcal{O}\left(\frac{B^2r_x}{\lambda N_e}\right), \end{aligned} \tag{4}$$

where $\mathbf{S} = \{\langle\mathcal{S}_1, \mathcal{T}_1\rangle, \cdots, \langle\mathcal{S}_{e_0-1}, \mathcal{T}_{e_0-1}\rangle, \langle\mathcal{S}_{e_0}, \mathcal{T}_{e_0}\rangle, \langle\mathcal{S}_{e_0+1}, \mathcal{T}_{e_0+1}\rangle, \cdots, \langle\mathcal{S}_{N_e}, \mathcal{T}_{N_e}\rangle\}$ denotes the full set of meta-samples, and $\mathbf{S}^{e_0} = \{\langle\mathcal{S}_1, \mathcal{T}_1\rangle, \cdots, \langle\mathcal{S}_{e_0-1}, \mathcal{T}_{e_0-1}\rangle, \langle\mathcal{S}_{e'_0}, \mathcal{T}_{e'_0}\rangle, \langle\mathcal{S}_{e_0+1}, \mathcal{T}_{e_0+1}\rangle, \cdots, \langle\mathcal{S}_{N_e}, \mathcal{T}_{N_e}\rangle\}$ represents the meta-samples with the e_0-th meta-example replaced as $\langle\mathcal{S}_{e'_0}, \mathcal{T}_{e'_0}\rangle$.

By generalizing S to be meta-samples $\mathbf{S}$ and h_S to be L2T $\mathbf{L}(\mathbf{S})$, we apply Corollary 21 in (Mohri & Rostamizadeh,

2010) to give the generalization bound of our algorithm $\mathbf{L}(\mathbf{S})$ in Theorem 2.

Theorem 2. *Let $\delta' = \delta - (N_e)^{\frac{1}{2(r+1)}-\frac{1}{4}}$ ($r > 1$ is required). Then for any sample $\mathbf{S}$ of size N_e drawn according to an algebraic β-mixing stationary distribution, and $\delta \geq 0$ such that $\delta' \geq 0$, the following generalization bound holds with probability at least $1 - \delta$:*

$$\left|R(\mathbf{L}(\mathbf{S})) - R_{N_e}(\mathbf{L}(\mathbf{S}))\right| < \mathcal{O}\left((N_e)^{\frac{1}{2(r+1)}-\frac{1}{4}}\sqrt{\log(\frac{1}{\delta'})}\right),$$

where $R(\mathbf{L}(\mathbf{S}))$ and $R_{N_e}(\mathbf{L}(\mathbf{S}))$ denote the expected risk and the empirical risk of L2T over meta-samples, respectively. A larger mixing parameter r, indicating more independence, would lead to a tighter bound.

Theorem 2 tells that as the number of transfer learning experiences, i.e., N_e, increases, L2T tends to produce a tighter generalization bound. This fact lays the foundation for further conducting L2T in an online manner which can gradually assimilate transfer learning experiences and continuously improve. The detailed proofs for Theorem 1 and 2 can be found in the supplementary.

5. Experiments

Datasets We evaluate the L2T framework on two image datasets, Caltech-256 (Griffin et al., 2007) and Sketches (Eitz et al., 2012). Caltech-256, collected from Google Images, contains a total of 30,607 images in 256 categories. The Sketches dataset, however, consists of 20,000 unique sketches by human beings that are evenly distributed over 250 different categories. We construct each pair of source and target domains by randomly sampling three categories from Caltech-256 as the source domain and randomly sampling three categories from Sketches as the target domain, which we give an example in the supplementary material. Consequently, there are $20,000/250 \times 3 = 720$ examples in a target domain of each pair. In total, we generate 1,000 training pairs for preparing transfer learning experiences, 500 validation pairs to determine hyperparameters of the reflection function, and 500 testing pairs to evaluate the reflection function. We characterize each image from both datasets with 4,096-dimensional features extracted by a convolutional neural network pre-trained by ImageNet.

In this paper we generate transfer learning experiences by ourselves, because we are the first to consider transfer learning experiences and there exists no off-the-shelf datasets. In real-world applications, either the number of labeled examples in a target domain or the transfer learning algorithm could vary from experience to experience. In order to mimic the real environment, we prepare each transfer learning experience by randomly selecting a transfer learning algorithm from a base set $\mathcal{A}$ and randomly setting the number of labeled target examples in the range of $[3, 120]$. The randomly generated training experiences, lying in the same environment (generated by one dataset), are non i.i.d., which fit the algebraical β-mixing assumption theoretically in Section 4.

Baselines and Evaluation Metrics We compare L2T with the following nine baseline algorithms in three classes:

- Non-transfer: **Original** builds a model using labeled data in a target domain only.
- Common latent space based transfer learning algorithms: **TCA** (Pan et al., 2011), **ITL** (Shi & Sha, 2012), **CMF** (Long et al., 2014), **LSDT** (Zhang et al., 2016), **STL** (Raina et al., 2007), **DIP** (Baktashmotlagh et al., 2013) and **SIE** (Baktashmotlagh et al., 2014).
- Manifold ensemble based algorithms: **GFK** (Gong et al., 2012).

The eight feature-based transfer learning algorithms also constitute the base set $\mathcal{A}$. Based on feature representations obtained by different algorithms, we use the nearest-neighbor classifier to perform three-class classification for the target domain.

One evaluation metric is classification accuracy on testing examples of a target domain. However, accuracies are incomparable for different target domains at different levels of difficulty. The other evaluation metric we adopt is the performance improvement ratio defined in Section 3.1, so as to compare the L2T over different pairs of domains.

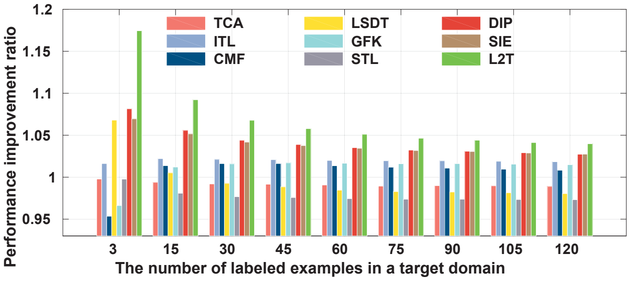
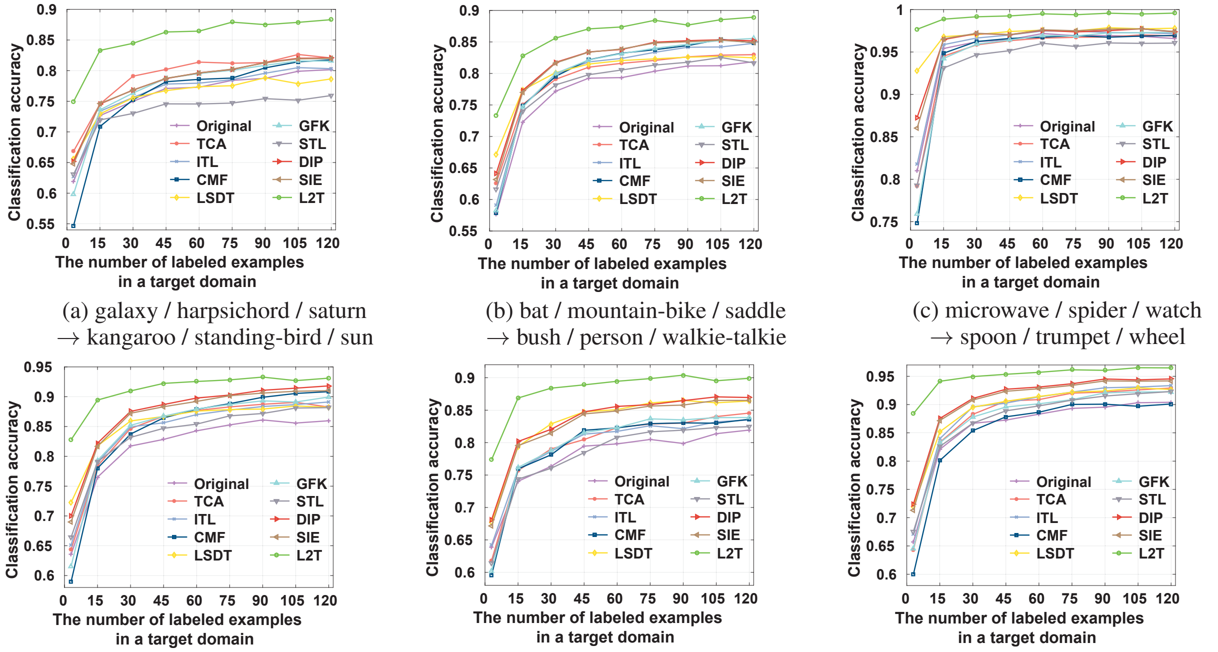


Figure 4. Average performance improvement ratio comparison over 500 testing pairs of source and target domains.

Performance Comparison In this experiment, we learn a reflection function from 1,000 transfer learning experiences, and evaluate the reflection function on 500 testing pairs of source and target domains by comparing the average performance improvement ratio to the baselines. In building the reflection function, we use 33 RBF kernels with the bandwidth δ_k in the range of $[2^{-8}\eta : 2^{0.5}\eta : 2^{8}\eta]$ where $\eta = \frac{1}{n_e^s n_e^t N_e}\sum_{e=1}^{N_e}\sum_{i,j=1}^{n_e^s,n_e^t}\|\mathbf{x}_{ei}^s\mathbf{W} - \mathbf{x}_{ej}^t\mathbf{W}\|_2^2$ follows the median trick (Gretton et al., 2012a). As Figure 4 shows, on average the proposed L2T framework outperforms the baselines up to 10% when varying the number of labeled samples in the target domain. As the number of labeled target examples increases from 3 to 120, the performance improvement ratio becomes smaller because the accuracy of Original without transfer tends to increase. The baseline



(a) galaxy / harpsichord / saturn → kangaroo / standing-bird / sun

(b) bat / mountain-bike / saddle → bush / person / walkie-talkie

(c) microwave / spider / watch → spoon / trumpet / wheel

(d) bridge / harp / traffic-light → door-handle / hand / present

(e) bridge / helicopter / tripod → key / parrot / traffic-light

(f) caculator / straw / french-horn → doorknob / palm-tree / scissors

Figure 3. Classification accuracies on six pairs of source and target domains.

algorithms behave differently. The transferable knowledge learned by LSDT helps a target domain a lot when training examples are scarce, while GFK performs poorly until training examples become more. STL is almost the worst baseline because it learns a dictionary from the source domain only but ignores the target domain. It runs at a high risk of failure especially when two domains are distant. DIP and SIE, which minimize the MMD and Hellinger distance between domains subject to manifold constraints, are competent. Note that we have run the paired t-test between L2T and each baseline with all the p-values in the order of 10^{-12}, concluding that the L2T is significantly superior.

We also randomly select six of the 500 testing pairs and compare classification accuracies by different algorithms for each pair in Figure 3. The performance of all baselines varies from pair to pair. Among all the baseline methods, TCA performs the best when transferring between domains in Figure 3a and LSDT is the most superior in Figure 3c. However, L2T consistently outperforms the baselines on all the settings. For some pairs, e.g., Figures 3a, 3c and 3f, the three classes in a target domain are comparably easy to tell apart, hence Original without transfer can achieve even better results than some transfer learning algorithms. In this case, L2T still improves by discovering the best transferable knowledge from the source domain, especially when the number of labeled examples is small (see Figure 3c and 3f). If two domains are very related, e.g., the source with "galaxy" and "saturn" and the target with "sun" in Figure 3a, L2T even finds out more transferable knowledge and contributes more significant improvement.

Varying the Experiences We further investigate how transfer learning experiences used to learn the reflection function influence the performance of L2T. In this experiment, we evaluate on 50 randomly sampled pairs out of the 500 testing pairs in order to efficiently investigate a wide range of cases in the following. The sampled set is unbiased and sufficient to characterize such influence, evidenced by the asymptotic consistency between the average performance improvement ratio on the 500 pairs in Figure 4 and that on the 50 pairs in the last line of Table 1. First, we fix the number of transfer learning experiences to be 1,000 and vary the set of base transfer learning algorithms. The results are shown in Table 1. Even with experiences generated by single base algorithm, e.g., ITL or DIP, the L2T can still learn a reflection function that significantly better (p-value < 0.05) decides what to transfer than using ITL or DIP directly. With more base algorithms involved, the transfer learning experiences are more diverse to cover more situations of source-target pairs and the knowledge transferred between them. As a result, the L2T learns a better reflection function and thereby achieves higher performance improvement ratios, which coincides with Theorem 2 where a larger r indicating more independence between experiences gives a tighter bound. Second, we fix the set of base algorithms to include all the eight baselines and vary the number of transfer learning experiences used for training. As shown in Figure 5, the average performance improvement ratio achieved by L2T tends to increase as the number of labeled examples in the target domain decreases, given that Original without transfer performs extremely poor with scarce labeled examples.

Table 1. The performance improvement ratios by varying different approaches used to generate transfer learning experiences. For example, "ITL+L2T" denotes the L2T learning from experiences generated by ITL only, and the second line of results for "ITL+L2T" is the p-value compared to ITL.

# of labeled examples	3	15	30	45	60	75	90	105	120
TCA	1.0181	1.0024	0.9965	0.9973	0.9941	0.9933	0.9938	0.9927	0.9928
ITL	1.0188	1.0248	1.0250	1.0254	1.0250	1.0224	1.0232	1.0224	1.0224
CMF	0.9607	1.0203	1.0224	1.0218	1.0190	1.0158	1.0144	1.0142	1.0125
LSDT	1.0828	1.0168	0.9988	0.9940	0.9895	0.9867	0.9854	0.9834	0.9837
GFK	0.9729	1.0180	1.0232	1.0243	1.0246	1.0219	1.0239	1.0229	1.0225
STL	0.9973	0.9771	0.9715	0.9713	0.9715	0.9694	0.9705	0.9693	0.9693
DIP	1.0875	1.0633	1.0518	1.0465	1.0425	1.0372	1.0365	1.0343	1.0317
SIE	1.0745	1.0579	1.0485	1.0448	1.0412	1.0359	1.0359	1.0334	1.0318
ITL + L2T	1.1210	1.0737	1.0577	1.0506	1.0456	1.0398	1.0394	1.0361	1.0359
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002	0.0002
DIP + L2T	1.1605	1.0927	1.0718	1.0620	1.0562	1.0500	1.0483	1.0461	1.0451
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
(LSDT/GFK /SIE) + L2T	1.1660	1.0973	1.0746	1.0652	1.0573	1.0506	1.0485	1.0451	1.0429
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
(TCA/ITL/CMF/GFK /LSDT/SIE/) + L2T	1.1712	1.0954	1.0707	1.0607	1.0529	1.0469	1.0449	1.0421	1.0416
	0.0000	0.0000	0.0001	0.0001	0.0106	0.0019	0.0002	0.0047	0.0106
all + L2T	1.1872	1.1054	1.0795	1.0699	1.0616	1.0551	1.0531	1.0500	1.0502

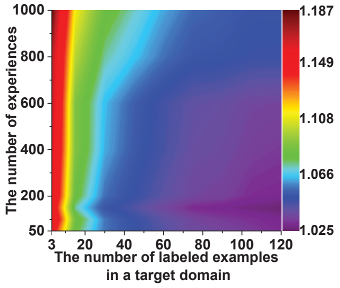


Figure 5. Varying the number of transfer learning experiences.

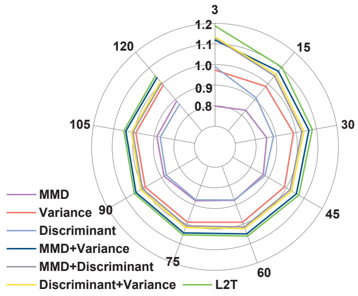


Figure 6. Varying the components constituted in the f.

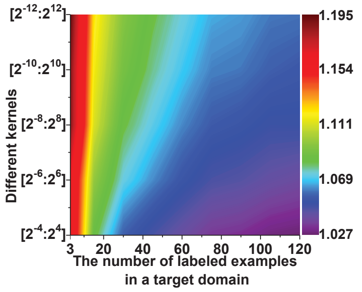


Figure 7. Varying the number of kernels considered in the f.

More importantly, it increases as the number of experiences increases, which coincides with Theorem 2.

Varying the Reflection Function We also study the influence of different configurations of the reflection function on the performance of L2T. First, we vary the components to be considered in building the reflection function f as shown in Figure 6. Considering single type, either MMD, variance, or the discriminant criterion, brings inferior performance and even negative transfer. L2T taking all the three factors into consideration outperforms the others, demonstrating that the three components are all necessary and mutually reinforcing. With all the three components included, we plot values of the learned $\boldsymbol{\beta}^*$ in the supplementary material. Second, we change the kernels used. In Figure 7, we present results by either narrowing down or extending the range $[2^{-8}\eta : 2^{0.5}\eta : 2^{8}\eta]$. Obviously, more kernels (e.g., $[2^{-12}\eta : 2^{0.5}\eta : 2^{12}\eta]$), capable of encrypting better transfer learning skills in the reflection function, achieve larger performance improvement ratios.

6. Conclusion

In this paper, we propose a novel L2T framework for transfer learning which automatically optimizes what and how to transfer between a source and a target domain by leveraging previous transfer learning experiences. In particular, L2T learns a reflection function mapping a pair of domains and the knowledge transferred between them to the performance improvement ratio. When a new pair of domains arrives, L2T optimizes what and how to transfer by maximizing the value of the learned reflection function. We believe that L2T opens a new door to improve transfer learning by leveraging transfer learning experiences. Many research issues, e.g., incorporating hierarchical latent feature factors as what to transfer and designing online L2T, can be further examined.

Acknowledgements

We thank the reviewers for their valuable comments to improve this paper. The research has been supported by National Grant Fundamental Research (973 Program) of China under Project 2014CB340304, Hong Kong CERG projects 16211214/16209715/16244616, Hong Kong ITF ITS/391/15FX and NSFC 61673202.

References

Argyriou, A., Evgeniou, T., and Pontil, M. Multi-task feature learning. In *NIPS*, pp. 41–48, 2007.

Baktashmotlagh, M., Harandi, M. T., Lovell, B. C., and Salzmann, M. Unsupervised domain adaptation by domain invariant projection. In *ICCV*, pp. 769–776, 2013.

Baktashmotlagh, M., Harandi, M. T., Lovell, B. C., and Salzmann, M. Domain adaptation on the statistical manifold. In *CVPR*, pp. 2481–2488, 2014.

Belmont, J. M., Butterfield, E. C., Ferretti, R. P., et al. To secure transfer of training instruct self-management skills. In Detterman, D. K. and Sternberg, R. J. P. (eds.), *How and How Much Can Intelligence be Increased*, pp. 147–154. Ablex Norwood, NJ, 1982.

Bernstein, S. Sur l'extension du théorème limite du calcul des probabilités aux sommes de quantités dépendantes. *Mathematische Annalen*, 97(1):1–59, 1927.

Blitzer, J., McDonald, R., and Pereira, F. Domain adaptation with structural correspondence learning. In *EMNLP*, pp. 120–128, 2006.

Cao, Q., Ying, Y., and Li, P. Similarity metric learning for face recognition. In *ICCV*, pp. 2408–2415, 2013.

Caruana, R. Multitask learning. *Machine Learning*, 28: 41–75, 1997.

Dai, W., Yang, Q., Xue, G.-R., and Yu, Y. Boosting for transfer learning. In *ICML*, pp. 193–200, 2007.

Eitz, M., Hays, J., and Alexa, M. How do humans sketch objects? *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31(4): 44:1–44:10, 2012.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pp. 1126–1135, 2017.

Gong, B., Shi, Y., Sha, F., and Grauman, K. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pp. 2066–2073, 2012.

Gopalan, R., Li, R., and Chellappa, R. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, pp. 999–1006, 2011.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *JMLR*, 13(Mar): 723–773, 2012a.

Gretton, A., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., Fukumizu, K., and Sriperumbudur, B. K. Optimal kernel choice for large-scale two-sample tests. In *NIPS*, pp. 1205–1213, 2012b.

Griffin, G., Holub, A., and Perona, P. Caltech-256 object category dataset. 2007.

Huber, P. J. et al. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

Long, M., Wang, J., Ding, G., Shen, D., and Yang, Q. Transfer learning with graph co-regularization. *TKDE*, 26(7): 1805–1818, 2014.

Long, M., Cao, Y., Wang, J., and Jordan, M. Learning transferable features with deep adaptation networks. In *ICML*, pp. 97–105, 2015.

Luria, A. R. *Cognitive Development: Its Cultural and Social Foundations*. Harvard University Press, 1976.

Maurer, A. Algorithmic stability and meta-learning. *JMLR*, 6(Jun):967–994, 2005.

Mo, K., Li, S., Zhang, Y., Li, J., and Yang, Q. Personalizing a dialogue system with transfer learning. *arXiv preprint arXiv:1610.02891*, 2016.

Mohri, M. and Rostamizadeh, A. Stability bounds for stationary φ-mixing and β-mixing processes. *JMLR*, 11 (Feb):789–814, 2010.

Pan, S. J. and Yang, Q. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2010.

Pan, S. J., Tsang, I. W., Kwok, J. T., and Yang, Q. Domain adaptation via transfer component analysis. *TNN*, 22(2): 199–210, 2011.

Pentina, A. and Lampert, C. H. Lifelong learning with non-iid tasks. In *NIPS*, pp. 1540–1548, 2015.

Raina, R., Battle, A., Lee, H., Packer, B., and Ng, A. Y. Self-taught learning: transfer learning from unlabeled data. In *ICML*, pp. 759–766, 2007.

Ruvolo, P. and Eaton, E. ELLA: An efficient lifelong learning algorithm. In *ICML*, pp. 507–515, 2013.

Shi, Y. and Sha, F. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. In *ICML*, pp. 1079–1086, 2012.

Thrun, S. and Pratt, L. *Learning to learn*. Springer Science & Business Media, 1998.

Tommasi, T., Orabona, F., and Caputo, B. Learning categories from few examples with multi model knowledge transfer. *TPAMI*, 36(5):928–941, 2014.

Tzeng, E., Hoffman, J., Darrell, T., and Saenko, K. Simultaneous deep transfer across domains and tasks. In *ICCV*, pp. 4068–4076, 2015.

Wei, Y., Zheng, Y., and Yang, Q. Transfer knowledge between cities. In *KDD*, pp. 1905–1914, 2016.

Yang, J., Yan, R., and Hauptmann, A. G. Adapting SVM classifiers to data with shifted distributions. In *ICDM*, pp. 69–76, 2007a.

Yang, J., Zhang, D., Yang, J.-y., and Niu, B. Globally maximizing, locally minimizing: unsupervised discriminant projection with applications to face and palm biometrics. *TPAMI*, 29(4), 2007b.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In *NIPS*, pp. 3320–3328, 2014.

Zhang, L., Zuo, W., and Zhang, D. LSDT: Latent sparse domain transfer learning for visual adaptation. *TIP*, 25 (3):1177–1191, 2016.