Towards Understanding Transfer Learning Algorithms Using Meta Transfer Features

Xin-Chun Li[1], De-Chuan Zhan[1(✉)], Jia-Qi Yang[1], Yi Shi[1], Cheng Hang[1], and Yi Lu[2]

[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210046, China
{lixc,zhandc,yangjq,shiy,hangc}@lamda.nju.edu.cn
[2] Huawei Technologies Co., Ltd., Nanjing 210012, China
luyi21@huawei.com

Abstract. Transfer learning, which aims to reuse knowledge in different domains, has achieved great success in many scenarios via minimizing domain discrepancy and enhancing feature discriminability. However, there are seldom practical determination methods for measuring the transferability among domains. In this paper, we bring forward a novel meta-transfer feature method (*MetaTrans*) for this problem. *MetaTrans* is used to train a model to predict performance improvement ratio from historical transfer learning experiences, and can consider both the *Transferability* between tasks and the *Discriminability* emphasized on targets. We apply this method to both shallow and deep transfer learning algorithms, providing a detail explanation for the success of specific transfer learning algorithms. From experimental studies, we find that different transfer learning algorithms have varying dominant factor deciding their success, so we propose a multi-task learning framework which can learn both common and specific experience from historical transfer learning results. The empirical investigations reveal that the knowledge obtained from historical experience can facilitate future transfer learning tasks.

Keywords: Transfer learning · Meta transfer features · Transferability · Discriminability

1 Introduction

In real-world tasks, test data usually differs from training data in the aspects of distributions, features, class categories, etc. Even there are some cases that the real applied circumstances occur in different domains without sufficient labels, i.e., in these cases, we need to exploit the full usage of the original model for adapting to the target domain, thus transfer learning is proposed.

Transfer learning algorithms can be grouped into two large categories according to using deep networks or not. The first category is shallow transfer learning, such as TCA [12], GFK [6], SA [4], KMM [8], ITL [15] and LSDT [22]. These algorithms can be further classified into instance-based and subspace-based ones

H. W. Lauw et al. (Eds.): PAKDD 2020, LNAI 12085, pp. 855–866, 2020.
https://doi.org/10.1007/978-3-030-47436-2_64

according to what to transfer [13]. In the category of deep transfer learning, discrepancy-based, adversarial-based, and reconstruction-based algorithms are the three main approaches [19], among which DAN [10] and RevGrad [5] are classical networks for transfer learning or domain adaptation[1].

Although many transfer learning algorithms are proposed, there are still few researches devoted to the three key issues in transfer learning, that is, when to transfer, how to transfer and what to transfer [13]. In this paper, we consider the three issues as one problem, i.e., we need to answer whether tasks can be transferred (when), and moreover, how to measure the *Transferability*. The later one implies the methods to transfer (how) and the information that can be transferred (what). As proposed in [3], we propose a novel *MetaTrans* method from both aspects of *Transferability* and *Discriminability*. *Transferability* means the similarity between the source and target domains, and *Discriminability* means how discriminative are the features extracted from a specific algorithm. In order to understand the internal mechanism of transfer learning algorithms and explain why they can improve the performance a lot, we extract some critical features according to these two dominant factors, which are called *Meta Transfer Features*.

Inspired by meta-learning methods [21] and the recent work [20], we build a model mapping *Meta Transfer Features* to the transfer performance improvement ratio using historical transfer learning experiences. Different from [20], we propose a multi-task learning framework to use historical experiences, with the reason that experiences from different algorithms vary a lot.

In this work, we make three contributions as follows:

- We propose a novel method *MetaTrans* to map *Meta Transfer Features* to the transfer performance improvement, from both aspects of *Transferability* and *Discriminability*.
- With the built mapping, we provide a detailed analysis of the success of both shallow and deep transfer algorithms.
- We propose a multi-task learning framework utilizing varying historical transfer experiences from different transfer learning algorithms as much as possible.

2 Related Works

In this section, we introduce some related works, including basic notations, theoretical analysis in transfer learning, deep domain adaptation and some recent researches.

2.1 Notations

In this work, we focus on the homogeneous unsupervised domain adaptation problem. The labeled source domain is denoted by $\mathcal{D}_S = \{\mathbf{X}_S, \mathbf{Y}_S\}$, and similarly, $\mathcal{D}_T = \{\mathbf{X}_T\}$ for the unlabeled target domain. In order to evaluate a specific

[1] In this paper, we do not focus on the difference between transfer learning and domain adaptation, we refer readers to [13] for details.

transfer learning algorithm, the real labels of target domain are denoted by $\mathbf{Y}_T$. We denote by $h \in \mathcal{H}$ the hypothesis (a.k.a. classifier in classification tasks) mapping from sample space $\mathcal{X}$ to label space $\mathcal{Y}$.

2.2 Theoretical Bound for Transfer Learning

From the previous theoretical result for domain adaptation [1], we have the generalization error bound on the target domain of a classifier trained in the source domain as follows:

Theorem 1. *Let $\mathcal{H}$ be a hypothesis space, and $\lambda = \min_{h\in\mathcal{H}}(\epsilon_S(h) + \epsilon_T(h))$ be the most ideal error of the hypothesis space on the source and target jointly, then for any $h \in \mathcal{H}$,*

$$\epsilon_T(h) \leq \epsilon_S(h) + d_{\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) + \lambda. \tag{1}$$

This bound contains three terms. The first one refers to the *Discriminability* of the features, being smaller if the learned features become more discriminative. The second one determines how similar are the source and target domains, the smaller the better, referred to as *Transferability*.

2.3 Deep Domain Adaptation

Deep domain adaptation contains adversarial-based and discrepancy-based methods. The framework of adversarial domain adaptation, such as RevGrad [5] and ADDA [18], utilizes the domain discriminator to separate the source and target domain as much as possible, that is, maximize the *Transferability* between domains. In addition, the task classifier component is used to maximize the performance of the source domain using the extracted features, in order to preserve the *Discriminability*. Similarly, discrepancy-based frameworks, such as DDC [17] and DAN [10], considering both the discrepancy loss (e.g. MMD loss) between two domains (*Transferability*) and the task specific loss (*Discriminability*).

2.4 Recent Researches

Recently, [3] analyzes the relation between *Transferability* and *Discriminability* in adversarial domain adaptation via the spectral analysis of feature representations, and proposed a batch spectral penalization algorithm to penalize the largest singular values to boost the feature discriminability. [20] proposes to use transfer learning experiences to automatically infer what and how to transfer in future tasks. [23] first addresses the gap between theories and algorithms, and then proposes new generalization bounds and a novel adversarial domain adaptation framework via the introduced margin disparity discrepancy.

3 MetaTrans Method

In this section, we introduce the proposed *MetaTrans*, including *Meta Transfer Features* and the multi-task learning framework.

3.1 Approximate Transferability

The *Transferability* refers to the discrepancy between two domains, and we can approximate it using different distance metrics. In this paper, we select the proxy $\mathcal{A}$-distance and the MMD distance as two approximations.

Proxy $\mathcal{A}$ Distance. The second term in the generalization bound in Eq. 1 is called the $\mathcal{H}$-divergence [9] between two domains. In order to approximate the $\mathcal{H}$-divergence with finite samples from source and target, the empirical $\mathcal{H}$-divergence is defined as

$$d_{\mathcal{H}}(D_S, D_T) = 2\left(1 - \min_{h \in \mathcal{H}} \left[\frac{1}{n_S} \sum_{\mathbf{x}:h(\mathbf{x})=0} I[\mathbf{x} \in D_S] + \frac{1}{n_T} \sum_{\mathbf{x}:h(\mathbf{x})=1} I[\mathbf{x} \in D_T]\right]\right), \tag{2}$$

where D_S and D_T are sets sampled from the corresponding marginal distribution with the size being n_S and n_T. $I[\cdot]$ is the identity function.

The empirical $\mathcal{H}$-divergence is also called proxy $\mathcal{A}$ distance. We can train a binary classifier h to discriminate the source and target domain, and the classification error can be used as an approximation of the proxy $\mathcal{A}$ distance,

$$d_{\mathcal{A}}(D_S, D_T) = 2(1 - 2err(h)), \tag{3}$$

where the $err(h)$ is the classification error of the specific classifier.

Maximum Mean Discrepancy. Another distance commonly used to measure the difference of two domains is MMD distance [7], a method to match higher-order moments of the domain distributions. The MMD distance is defined as

$$d_{mmd} = \left\| E_{\mathbf{x} \in \mathcal{D}_S}\left[\phi(\mathbf{x})\right] - E_{\mathbf{x} \in \mathcal{D}_T}\left[\phi(\mathbf{x})\right] \right\|_{\mathcal{H}}, \tag{4}$$

where ϕ is a function maps the sample to the reproducing kernel Hilbert space $\mathcal{H}$. In order to approximate the MMD distance from finite samples, the empirical MMD distance is defined as

$$d_{mmd} = \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \phi(\mathbf{x}_i) - \frac{1}{n_T} \sum_{j=1}^{n_T} \phi(\mathbf{x}_j) \right\|_{\mathcal{H}}. \tag{5}$$

In order to get the empirical MMD distance, a kernel function is needed, and the commonly used kernel is the RBF kernel defined as $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{\sigma^2}\right)$. To avoid the trouble of selecting the best kernel bandwidth σ, we use multi-kernel MMD (MK-MMD), and the multi-kernel is defined as a linear combination of N RBF kernels with the form $\mathcal{K} = \sum_{k=1}^{N} \mathcal{K}_k$.

3.2 Approximate Discriminability

The *Discriminability* measures the discriminative ability of feature representations. We propose three approximate features including the empirical source error, the supervised discriminant criterion and the unsupervised discriminant criterion.

Source Domain Error. In the generalization bound for domain adaptation (Eq. 1), the source error is an important factor determining the target generalization error. The empirical source error is defined as

$$\epsilon_S(h) = \frac{1}{n_S} \sum_{i=1}^{n_S} l(h(\mathbf{x}_i), y_i), \tag{6}$$

where y_i is the real label for the i-th sample and l is the loss function.

Supervised Discriminant Criterion. According to the supervised dimension reduction methods (such as LDA), the ratio of between-class scatter and inner-class scatter implies the discriminative level of the features.

Supposing there are C classes in the source domain, and the mean vector for these classes are $\{\mu_c\}_{c=1}^{C}$ accordingly, then we have the inner-class scatter as

$$d_{inner} = \frac{1}{n_S} \sum_{c=1}^{C} \sum_{j=1}^{n_c} \|\mathbf{x}_{cj} - \mu_c\|_2^2 , \tag{7}$$

where the c-th class has n_c samples and $\mathbf{x}_{cj}$ is the j-th sample of the c-th class. Meanwhile, the between-class scatter is defined as

$$d_{between} = \frac{1}{n_S} \sum_{c=1}^{C} n_c \|\mu_c - \mu_0\|_2^2 , \tag{8}$$

where μ_0 is the mean center of all samples in the source domain. We approximate the source discriminability with the formulation

$$c_{sdc} = \frac{d_{between}}{d_{inner} + d_{between}} \tag{9}$$

where c_{sdc} is the notation of supervised discriminant criterion.

Unsupervised Discriminant Criterion. If no labeled data can be obtained, the supervised discriminant criterion can not be used. Towards measuring the discriminant ability of the feature representations in the target domain with no label, the unsupervised discriminant criterion can be applied. Similarly, there are two types of scatter in unsupervised discriminant criterion called the local-scatter and global-scatter.

The local-scatter is defined as

$$d_{local} = \frac{1}{n_T^2} \sum_{i=1}^{n_T} \sum_{j=1}^{n_T} \mathbf{H}_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 , \tag{10}$$

where $\mathbf{H}$ is defined as neighbor affinity matrix, being $\mathbf{K}_{ij}$ when $\mathbf{x}_i$ and $\mathbf{x}_j$ are neighbors to each other, and being 0 otherwise. $\mathbf{K}_{ij}$ is the kernel matrix item

using the multi-kernel proposed as before. And similarly, the global scatter is defined as

$$d_{global} = \frac{1}{n_T^2} \sum_{i=1}^{n_T} \sum_{j=1}^{n_T} (\mathbf{K}_{ij} - \mathbf{H}_{ij}) \left\| \mathbf{x}_i - \mathbf{x}_j \right\|_2^2 . \tag{11}$$

Therefore, we use the ratio of the global scatter in the total scatter as an approximation to the discriminability of the feature representations in the target domain, which is defined as

$$c_{udc} = \frac{d_{global}}{d_{local} + d_{global}}, \tag{12}$$

and the c_{udc} is the abbreviation of unsupervised discriminant criterion.

3.3 Problem Statements

With the above approximations, the *Meta Transfer Features* are denoted as a five-tuple $(d_{\mathcal{A}}, d_{mmd}, \epsilon_S, c_{sdc}, c_{udc})$. In transfer learning, we always focus on the performance improvement ratio brought by using a specific transfer learning algorithm compared to the case without using it. We build a machine learning model in source domain $\mathcal{D}_S = \{\mathbf{X}_S, \mathbf{Y}_S\}$, and we denote it as h_S. Without using any transfer learning algorithms, the target domain error is defined as $\epsilon_{wo} = \frac{1}{n_T} \sum_{i=1}^{n_T} l(h_S(\mathbf{X}_{Ti}), \mathbf{Y}_{Ti})$, where l is the loss function and $\mathbf{X}_{Ti}$ is the i-th sample in target domain. A specific transfer learning algorithm g, with the input as $\mathbf{X}_S, \mathbf{X}_T$, could output the aligned data samples as $\hat{\mathbf{X}}_S, \hat{\mathbf{X}}_T$[2]. The aligned source and target domains become $\{\hat{\mathbf{X}}_S, \mathbf{Y}_S\}$ and $\{\hat{\mathbf{X}}_T\}$, and then similarly, we can get the new target domain error $\epsilon_w = \frac{1}{n_T} \sum_{i=1}^{n_T} l(\hat{h}_S(\hat{\mathbf{X}}_{Ti}), \mathbf{Y}_{Ti})$, where $\hat{h}_S$ is the model learned from new source domain samples. If ϵ_w is smaller than ϵ_{wo}, we believe that g has made an improvement, and the ratio is defined as r_{imp}:

$$r_{imp} = \frac{\epsilon_{wo} - \epsilon_w}{\epsilon_{wo}} \tag{13}$$

Given the source and target domains $\mathcal{D}_S = \{\mathbf{X}_S, \mathbf{Y}_S\}$ and $\mathcal{D}_T = \{\mathbf{X}_T\}$, using a transfer learning algorithm g, we can get representations $\hat{\mathcal{D}}_S = \{\hat{\mathbf{X}}_S, \mathbf{Y}_S\}$ and $\hat{\mathcal{D}}_T = \{\hat{\mathbf{X}}_T\}$. From $\mathcal{D}_S$ and $\mathcal{D}_T$, we can get a five tuple *Meta Transfer Features* denoted as $(d_{\mathcal{A}}, d_{mmd}, \epsilon_S, c_{sdc}, c_{udc})$, and similarly, from $\hat{\mathcal{D}}_S$ and $\hat{\mathcal{D}}_T$, we can get another five tuple denoted as $\left(\hat{d}_{\mathcal{A}}, \hat{d}_{mmd}, \hat{\epsilon}_S, \hat{c}_{sdc}, \hat{c}_{udc}\right)$. We combine this two tuples together, and get the features denoted as $\mathbf{x}^{meta}$. Using these features, we want to regress the transfer improvement ratio r_{imp} denoted as $\mathbf{y}^{meta}$.

From historical transfer learning experiences, we can get pairs of ($\mathbf{x}^{meta}$, $\mathbf{y}^{meta}$), and then we can build a model maps *Meta Transfer Features* to the transfer improvement ratio. With this obtained model, we can have a better understanding of the internal mechanism of transfer learning algorithms and provide some prior knowledge for future transfer learning tasks.

[2] We only give the most common case, some algorithms like instance-based ones will output a group of weights, and we can apply importance sampling to get new source domain samples.

3.4 Multi-task Learning Framework

Considering transfer learning algorithms are designed with different mechanisms, it is not wise to build a single mapping from their experiences, losing the specialities. Additionally, we want to learn something common which can be applied to new transfer learning algorithms so that we can not train models individually. Therefore, we propose a multi-task learning framework to learn common and specific knowledge from varying transfer learning experiences.

To be specific, given the transfer learning experiences of T different algorithms denoted as $\{\{(\mathbf{x}_{ti}^{meta}, \mathbf{y}_{ti}^{meta})\}_{i=1}^{N_t}\}_{t=1}^{T}$. For simplicity, we use linear regression with regularization as our mapping function. We divide mapping functions into two parts, the common and specific ones, denoted by $(\mathbf{w}, b)$ and $\{(\mathbf{w}_t, b_t)\}_{t=1}^{T}$ correspondingly. Then our optimization target is:

$$\min_{\theta} L = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N_t}\left((\mathbf{w}+\mathbf{w}_t)^T\mathbf{x}_{ti}^{meta} + b + b_t - \mathbf{y}_{ti}^{meta}\right)^2 + \lambda R(\mathbf{w}, \{\mathbf{w}_t\}_{t=1}^{T}), \tag{14}$$

where $R(\mathbf{w}, \{\mathbf{w}_t\}_{t=1}^{T})$ is the regularization term, such as the $L2$-norm regularization and $\theta = \{\mathbf{w}, b, \{\mathbf{w}\}_{t=1}^{T}, \{b\}_{t=1}^{T}\}$ denotes the parameters to be learned. In order to solve this problem, we use the alternative optimization strategy. First, we fix the global parameters $(\mathbf{w}, b)$ and optimize $(\mathbf{w}_t, b_t)$ for each task, and then we fix local parameters $(\mathbf{w}_t, b_t)_{t=1}^{T}$ and optimize the $(\mathbf{w}, b)$ alternatively.

4 Experimental Studies

In this section, we display some experiments with both synthetic and public data.

4.1 Understanding Meta Transfer Features

One of the contributions of this work is the proposed *Meta Transfer Features*, so we will provide some experimental results on synthetic data to understand why these features matter so much.

In order to understand the *Transferability*, we sample data from two 2-dim gaussian distributions as the source and target domain, which is shown in the top row of Fig. 1. From the figure, the proxy $\mathcal{A}$ distance (HDIV in figure) and MMD distance become larger when two domains become further. As to the *Discriminability*, we sample data from five gaussian distributions as five classes. From the bottom row in Fig. 1, it is shown that both the supervised and unsupervised discriminative criterion become larger with the overlap among classes becomes smaller, which means the features are more discriminative for classification.

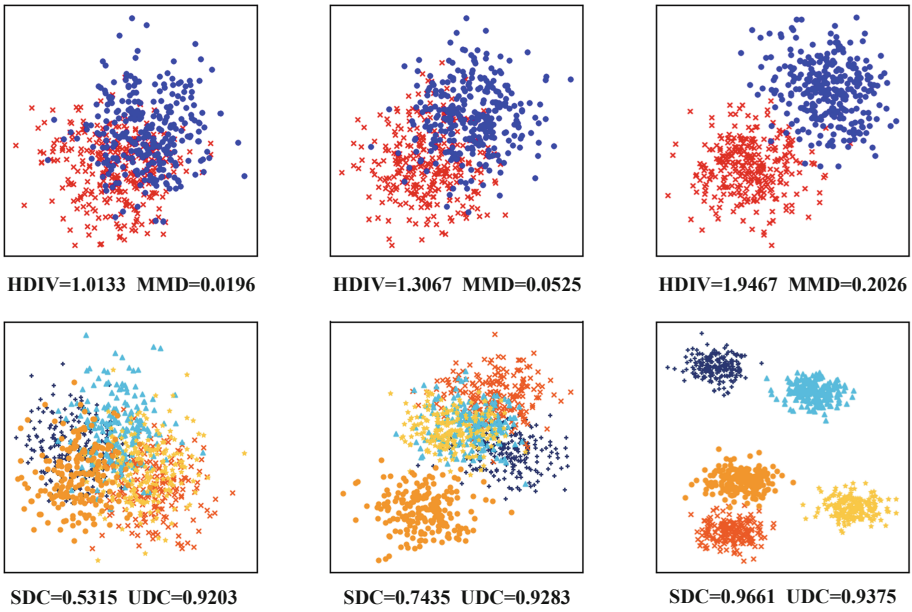


Fig. 1. Understanding *Meta Transfer Features*. The first row illustrates the *Transferability* between source and target domains, while the second row shows the *Discriminability* of features with five classes.

4.2 Understanding Transfer Learning Methods

As proposed further, different transfer learning algorithms have their individual mechanisms, so we will provide experimental results for this finding.

Shallow Transfer Methods. In this section, we implement TCA [12], SA [4] and ITL [15] as examples, showing the different mechanisms among them.

In order to visualize the learned representations, we use synthetic data constructed as follows: we sample data from two 2-dim gaussian distributions as two classes in source domain (S0, S1 in Fig. 2 (a)), and then we rotate the guassian means with a definite angle, and the new means are used to sample target data (T0, T1 in Fig. 2 (a)) with the same covariance. Then we use TCA, SA and ITL to get aligned distributions in 1-dim space, and for every algorithm, we select the best parameters to get almost the same 10% improvement in classification accuracy compared to the case without using this algorithm. Considering the overlap between two classes in two domains in 1-dim space, we plot them separately with different y-axis values as in Fig. 2. From this visualization result, it is obvious that ITL can get a more discriminative representation then TCA and SA, for the appearance that the samples in different classes are largely separated as shown in Fig. 2 (d). The result fits well with the information-theoretic factors considered in the designation process of ITL, and we refer readers to [15] for

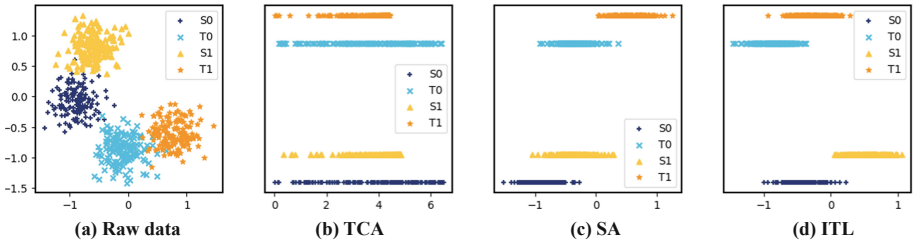


Fig. 2. Understanding shallow transfer learning algorithms. From left to right: (a) The synthetic data. (b) The 1-dim features obtained from TCA. (c) The 1-dim features obtained from SA. (d) The 1-dim features obtained from ITL.

more details. In addition, TCA can get a better alignment between source and target domains as shown in Fig. 2 (b).

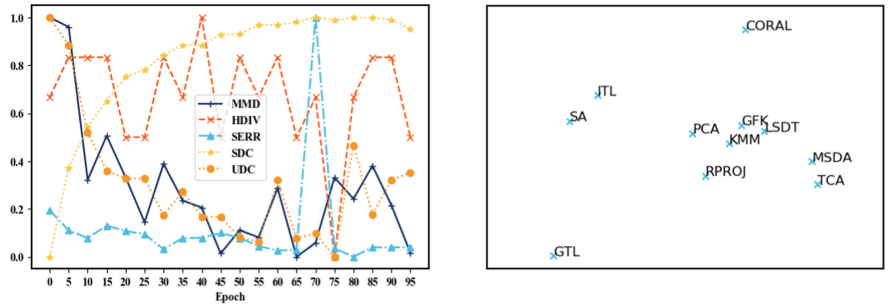
Deep Transfer Methods. Aside from the shallow transfer learning algorithms, we explore the change of *Meta Transfer Features* in the learning process of deep transfer learning algorithms. We take DAN [10] as an example. We use the Amazon (**A**) and DSLR (**D**) in Office [14] dataset as source and target domains. For each training epoch, we extract *Meta Transfer Features* from the hidden representations learned from DAN network, and we plot the change of these features as shown in Fig. 3 (a) (the plot is normalized with min-max normalizetion). It is obvious that MMD distance (MMD in Figure) becomes smaller and smaller with the optimization process of domain alignment mechanism in DAN, while proxy $\mathcal{A}$ distance (HDIV in Figure) oscillates a lot. In addition, the *sdc* becomes smaller, showing that features could be more confusing with the overlap between two domains becoming larger.

4.3 Prediction Results

Transfer learning experiences are constructed from sub-tasks sampled from the classical datasets: Office [14], Caltech [6], MNIST (**M**) and USPS (**U**). The Office and Caltech datasets have four domains in total: Amazon (**A**), Caltech (**C**), DSLR (**D**) and Webcam (**W**). For a specific source and target combination such as **A** → **C**, we sample tasks with a subset classes in the total 10 classes. For example, we can sample a 4-classes classification task, and there are will be 210 unique tasks in total can be sampled.

For the prediction experiments, we only focus on shallow transfer learning algorithms, including RPROJ[3], PCA, TCA [12], MSDA [2], CORAL [16], GFK [6], ITL [15], LSDT [22], GTL [11] and KMM [8]. These algorithms contain almost all kinds of shallow transfer learning algorithms, such as instance-based, subspace-based, manifold-based, information-based and reconstruction-based.

[3] Dimensional reduction with Random Projection.



(a) Meta Transfer Features in DAN

(b) Task (Transfer Algorithms) Visualization

Fig. 3. (a) Understanding deep transfer learning algorithms: the change of *Meta Transfer Features* in the training process. (b) Task visualization using MDS, mapping the learned weights into the 2-dim space.

For each sampled task, we apply all of these algorithms with random selected hyperparameters and get the $(\mathbf{x}^{meta}, \mathbf{y}^{meta})$ pairs.

We compare our proposed multi-task learning framework (Meta-MTL) with two baselines: the first one is training a single model together (Meta-Sin), and the second one is training a model for each transfer algorithm individually (Meta-Ind). We use both MSE and MAE as the evaluation criterions. The prediction results can be found in Table 1, which verifies the validity of our MTL framework. Our MTL framework can predict the transfer improvement ratio more accurate for unseen transfer tasks. It also explains that experiences from different transfer learning algorithms should not be utilized equally. The first column displays the source and domain pairs we use to obtain transfer learning experiences, and we find the ignored dataset information also matters a lot, which will be the future work to research.

Table 1. Prediction results of different methods of utilizing the transfer learning experiences.

Train and test sets	Method	MSE	MAE
Train: $\mathbf{A} \rightarrow \mathbf{C}, \mathbf{A} \rightarrow \mathbf{D}, \cdots, \mathbf{W} \rightarrow \mathbf{D}$	Meta-Sin	0.0339	0.1573
Test: $\mathbf{U} \rightarrow \mathbf{M}, \mathbf{M} \rightarrow \mathbf{U}$	Meta-Inv	0.0418	0.1724
	Meta-MTL	**0.0314**	**0.1507**
Train: $\mathbf{A} \rightarrow \mathbf{C}$	Meta-Sin	0.0104	0.0821
Test: $\mathbf{A} \rightarrow \mathbf{D}$	Meta-Inv	0.0162	0.1065
	Meta-MTL	**0.0081**	**0.0729**

In addition, in order to visualize the difference among transfer learning algorithms, we use MDS to get the lower representations in 2-dim space keeping the euclidean distances among their specific weights unchanged as much as possible. We plot the relationships in Fig. 3 (b). From this figure, we can find and search some similar transfer learning methods for alternative algorithms, and meanwhile, some diverse algorithms can be used for ensemble learning. To be specific, we find MSDA and TCA may be alternative transfer learning methods in this experiment.

5 Conclusion

In this paper, we propose *MetaTrans* from both *Transferability* and *Discriminability* aspects and give a comprehensive understanding of both shallow and deep transfer learning algorithms. As to the use of historical transfer learning experiences, we propose a multi-task learning framework, and the experimental results show that it could utilize experiences better and predict future transfer performance improvement more accurate. Considering more meta-features, taking the dataset information into consideration or learning task embeddings are future works.

Acknowledgments. This research was supported by National Key R&D Program of China (2018YFB1004300), NSFC (61773198, 61632004, 61751306), NSFC-NRF Joint Research Project under Grant 61861146001, and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

1. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: Advances in Neural Information Processing Systems, pp. 137–144 (2007)
2. Chen, M., Xu, Z., Weinberger, K.Q., Sha, F.: Marginalized denoising autoencoders for domain adaptation. In: Proceedings of the 29th International Conference on Machine Learning, pp. 1627–1634 (2012)
3. Chen, X., Wang, S., Long, M., Wang, J.: Transferability vs. discriminability: batch spectral penalization for adversarial domain adaptation. In: Proceedings of the 36th International Conference on Machine Learning, pp. 1081–1090 (2019)
4. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2960–2967 (2013)
5. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: Proceedings of the 32nd International Conference on Machine Learning, pp. 1180–1189 (2015)
6. Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2066–2073 (2012)

7. Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.J.: A kernel method for the two-sample-problem. In: Advances in Neural Information Processing Systems, pp. 513–520 (2007)
8. Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., Smola, A.J.: Correcting sample selection bias by unlabeled data. In: Advances in Neural Information Processing Systems, pp. 601–608 (2007)
9. Kifer, D., Ben-David, S., Gehrke, J.: Detecting change in data streams. In: Proceedings of the 30th International Conference on Very Large Data Bases, pp. 180–191 (2004)
10. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: Proceedings of the 32nd International Conference on Machine Learning, pp. 97–105 (2015)
11. Long, M., Wang, J., Ding, G., Shen, D., Yang, Q.: Transfer learning with graph co-regularization. IEEE Trans. Knowl. Data Eng. **26**, 1805–1818 (2013)
12. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE Trans. Neural Netw. **22**, 199–210 (2010)
13. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**, 1345–1359 (2009)
14. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 213–226. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_16
15. Shi, Y., Sha, F.: Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. In: Proceedings of the 29th International Conference on International Conference on Machine Learning, pp. 1275–1282 (2012)
16. Sun, B., Feng, J., Saenko, K.: Return of frustratingly easy domain adaptation. In: AAAI Conference on Artificial Intelligence (2016)
17. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: maximizing for domain invariance. arXiv preprint arXiv:1412.3474 (2014)
18. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7167–7176 (2017)
19. Wang, M., Deng, W.: Deep visual domain adaptation: a survey. Neurocomputing **312**, 135–153 (2018)
20. Wei, Y., Zhang, Y., Huang, J., Yang, Q.: Transfer learning via learning to transfer. In: Proceedings of the 35th International Conference on Machine Learning, pp. 5085–5094 (2018)
21. Ye, H.J., Hu, H., Zhan, D.C., Sha, F.: Learning embedding adaptation for few-shot learning. arXiv preprint arXiv:1812.03664 (2018)
22. Zhang, L., Zuo, W., Zhang, D.: LSDT: latent sparse domain transfer learning for visual adaptation. IEEE Trans. Image Process. **25**, 1177–1191 (2016)
23. Zhang, Y., Liu, T., Long, M., Jordan, M.: Bridging theory and algorithm for domain adaptation. In: Proceedings of the 36th International Conference on Machine Learning, pp. 7404–7413 (2019)