Distant Transfer Learning via Deep Random Walk

Qiao Xiao[1] and Yu Zhang[1,2,*]

[1]Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China
[2]Peng Cheng Laboratory, Shenzhen, China
xiaoq3@mail.sustech.edu.cn, yu.zhang.ust@gmail.com

Abstract

Transfer learning, which is to improve the learning performance in the target domain by leveraging useful knowledge from the source domain, often requires that those two domains are very close, which limits its application scope. Recently, distant transfer learning has been studied to transfer knowledge between two distant or even totally unrelated domains via unlabeled auxiliary domains that act as a bridge in the spirit of human transitive inference that two completely unrelated concepts can be connected through gradual knowledge transfer. In this paper, we study distant transfer learning by proposing a DeEp Random Walk basEd distaNt Transfer (DERWENT) method. Different from existing distant transfer learning models that implicitly identify the path of knowledge transfer between the source and target instances through auxiliary instances, the proposed DERWENT model can explicitly learn such paths via the deep random walk technique. Specifically, based on sequences identified by the random walk technique on a data graph where source and target data have no direct connection, the proposed DERWENT model enforces adjacent data points in a sequence to be similar, makes the ending data point be represented by other data points in the same sequence, and considers weighted classification losses of source data. Empirical studies on several benchmark datasets demonstrate that the proposed DERWENT algorithm yields the state-of-the-art performance.

Introduction

Transfer learning (Yang et al. 2020) aims to effectively enhance the performance of the target domain by learning useful knowledge from the source domain and it has a wide range of applications (Zhang et al. 2019; Uribe 2010; Pan et al. 2011), especially when the target domain has limited or no label information. Using a large number of labeled data in the source domain to improve the performance in the target domain with limited or even no labeled training data via transfer learning models can greatly reduce the cost of labeling in the target domain.

A major assumption of traditional transfer learning is that the source and target domains should be close or similar to each other. When there is a large discrepancy between the target domain and the source domain, traditional transfer learning methods likely fail to work and even lead to the 'negative transfer' phenomenon (Pan and Yang 2010; Yang et al. 2020). Instead, distant transfer learning (Tan et al. 2015, 2017) is proposed to handle this situation. Inspired by the transitive learning ability of human that two unrelated concepts can be connected via some intermediate concepts as a bridge, distant transfer learning uses data in auxiliary domains as such bridge to connect two distant domains, which makes the knowledge transfer between two distant domains possible. Distant transfer learning broadens the application scope of transfer learning and makes the learning system close towards human intelligence.

As pioneered in distant transfer learning, Tan et al. (2015) require that auxiliary domains include both characteristics of the source and target domains in a form of the co-occurrence data and propose a matrix-factorization-based model to achieve one-step transitive learning through the auxiliary domain. By relaxing such requirement on the data form in the auxiliary domain, the Distant Domain Transfer Learning (DDTL) method (Tan et al. 2017) utilizes the idea of self-paced learning (Kumar, Packer, and Koller 2010) to select both useful source and auxiliary data based on the reconstruction error for improving the performance of the target domain which has limited labeled data. However, those two studies cannot explicitly identify the transfer paths between the source and target domains via auxiliary domains.

In this paper, we follow the setting of (Tan et al. 2017) to study distant transfer learning with an objective to identify transfer paths between two distant domains, which is what previous studies cannot achieve. An advantage of identifying the transfer paths is to improve the interpretability of the model by visualizing the transfer process. To achieve that, we adopt deep random walk to generate transfer paths between those two domains. Specifically, as shown in Figure 1, we construct a graph on all the data from all the domains with edge weights measuring the similarities of pairs of data based on the hidden feature representation learned from a neural network. Note that there are no edges between source and target data in the graph as directly transferring is not so feasible. Then based on the constructed graph, we can generate sequences to connect source and target data through auxiliary data. For each sequence, the proposed DeEp Random Walk basEd distaNt Transfer (DERWENT) model enforces adjacent data points in this sequence to be similar and makes
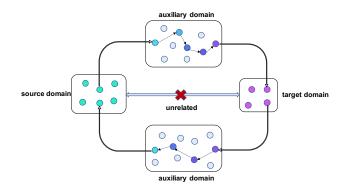
Figure 1: An illustration of distant transfer learning. In distant transfer learning, the source and target domains cannot be directly transferred since the discrepancy between domains is too large, making directly transfer fail to work. The proposed DERWENT method can automatically find the transfer paths via deep random walk between the source domain and the target domain through auxiliary domains.

the ending data point in this sequence be represented by other data points via a Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997). Moreover, the DER-WENT model considers to minimize the weighted classification loss of source data in a sequence with weights depending on the data similarities.

In summary, the main contributions of this paper are three-fold.
1. We propose a novel DERWENT model for distant transfer learning by utilizing deep random walk to generate transfer paths between the source and target domains.
2. We conduct extensive experiments on twenty distant transfer learning tasks constructed from several benchmark datasets to validate the effectiveness of the proposed DERWENT model.
3. The proposed DERWENT model can identify transfer paths, which can improve the model interpretability by visualizing such sequences.

Related Works

In transfer learning, we are usually given a source domain $\mathcal{D}_s$ and a target domain $\mathcal{D}_t$. There are mainly three typical types of transfer learning algorithms, including instance-based transfer learning (Dai et al. 2007; Khan and Heisterkamp 2016), feature-based transfer learning (Pan et al. 2009; Hu and Yang 2011), and parameter-based transfer learning (Pan et al. 2008). Recently, transfer learning is mostly combined with deep neural networks (Tzeng et al. 2015; Long et al. 2016). However, the aforementioned works study traditional transfer learning which requires that the source and target domains are close and they may not achieve good performance under the distant transfer learning setting or even lead to negative transfer problem.

For distant transfer learning, Tan et al. (2015) use an auxiliary domain as a bridge between distant domains. However, the data in the auxiliary domain need to take the form of the co-occurrence data and the learning model relies on matrix

factorization, which greatly limits its application scope. The major differences of the proposed DERWENT method with it are that the DERWENT method has no such requirement on the data form in auxiliary domains and that the DERWENT method can achieve a multi-step transition in the auxiliary domains while (Tan et al. 2015) can take only one step. DDTL (Tan et al. 2017) aims to select useful data from the source and auxiliary domains through a selective learning method inspired by self-paced learning to improve the performance in the target domain. The proposed DERWENT model is different from DDTL in mainly two aspects. Firstly, DDTL selects source and auxiliary data based on the idea of self-paced learning but it cannot explicitly give the transfer paths between the source and target domains, while the proposed DERWENT method can do that with the help of the deep random walk. Secondly, DDTL selects useful data from the source and auxiliary domains according to the reconstruction error, while the proposed DERWENT method relies on the deep random walk with two designed criteria including the similarity between adjacent data in a sequence sampled in the deep random walk and the reconstruction of the ending data point in the sequence based on other data points.

As mentioned by Tan et al. (2017), Self-Taught Learning (STL) (Raina et al. 2007), which aims to learn a good feature representation through a large amount of unlabeled data, can also work under the distant transfer learning setting where the auxiliary data take the role of unlabeled data. STL is an unsupervised method with the original formulation relying on linear sparse coding models. In recent years, with the development of deep learning, STL has adopted deep neural networks as basic models and has achieve better performance as shown in (Kemker and Kanan 2017; Gan et al. 2014). However, distant transfer learning is different from STL in that STL treats the unlabeled data as the source domain, while such unlabeled data is treated as data in the auxiliary domains for distant transfer learning.

DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) aims to obtain sequences of data nodes in a graph for model training. DeepWalk mainly uses random walk to sample sequences from the graph, and it is to maximize the co-occurrence probability among data nodes that appear within a window in the spirit of the SkipGram model (Mikolov et al. 2013). Different from Deepwalk, the proposed DERWENT method uses random walk for a data graph to generate the transfer paths between the source and target domains through auxiliary domains for distant transfer learning and it is to maximize the similarity between adjacent data points in a sampled sequence and to minimize the reconstruction error of the ending data point with respect to other data points in the same sequence via the LSTM.

Tan et al. (2014) and Ng, Wu, and Ye (2012) use random walk to transfer information between two heterogeneous domains. Those two works are different from ours in two aspects. Firstly, the problem settings are different. Those two works require the existence of co-occurrence data for transferring between two heterogeneous domains, while our work has no such requirement. Secondly, the ways to use random walk are different. Those two works use random walk to compute the probabilities of traversing between source and target

instances and then use such probabilities to do the transfer in terms of instances or features, however, our work uses random walk for sampling sequences to connect two domains through auxiliary domains and then uses sampled sequences to update the entire network based on three proposed losses.

The DERWENT Model

In this section, we introduce the proposed DERWENT model.

Problem Settings

By following the problem setting in DDTL (Tan et al. 2017) where there are a source domain and a target domain, the source domain has a large labeled training dataset $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$, where $y_i^s \in \{0, 1\}$ is the class label of the ith data point $\mathbf{x}_i^s$ in the source domain and n_s denotes the number of data points in the source domain, and the target domain has a small labeled training dataset $\mathcal{D}_t = \{(\mathbf{x}_i^t, y_i^t)\}_{i=1}^{n_t}$, where $y_i^t \in \{0, 1\}$ is the class label of the ith data point $\mathbf{x}_i^t$ in the target domain and n_t denotes the number of data points in the target domain. Here we assume $n_s \gg n_t$. Since the source and target domains have a large discrepancy, the direct transfer from the source domain to the target domain may have no effect or even negative effect to improve the performance of the target domain. Instead, we assume that there is an auxiliary unlabeled dataset $\mathcal{D}_a = \{\mathbf{x}_1^a, \ldots, \mathbf{x}_{n_a}^a\}$ where $n_a \gg n_t$ denotes the number of data points. This auxiliary dataset contains data points from diverse auxiliary domains and it will act as a bridge to help transfer the knowledge from the source domain to the target domain in order to improve the performance of the target domain.

The Model

To achieve that, we propose the DERWENT model which is based on deep random walk.

In the DERWENT model, we first learn a hidden representation for all the data in the source, auxiliary, and target domains as $\hat{\mathbf{x}}_i^* = \phi(\mathbf{x}_i^*)$, where $\phi(\cdot)$ denotes a feature extraction network and the superscript $*$ in $\mathbf{x}_i^*$ can be s, a, or t.

To measure similarities between data points, we construct a graph G on all the data from all the domains with each data point corresponding to a node in this graph and edge weights defined as

$$
\begin{aligned}
e(\hat{\mathbf{x}}, \hat{\mathbf{x}}) &= 0 \quad \forall \mathbf{x} \in \mathcal{D}_s \cup \mathcal{D}_a \cup \mathcal{D}_t \\
e(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2) &= \exp\{\cos(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2)\} \quad \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D}_* \, \forall * \in \{s, a, t\} \\
e(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2) &= \exp\{\eta_1 \cos(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2)\} \quad \forall \mathbf{x}_1 \in \mathcal{D}_s \, \mathbf{x}_2 \in \mathcal{D}_a \\
e(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2) &= \exp\{\eta_1 \cos(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2)\} \quad \forall \mathbf{x}_1 \in \mathcal{D}_a \, \mathbf{x}_2 \in \mathcal{D}_s \\
e(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2) &= \exp\{\eta_2 \cos(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2)\} \quad \forall \mathbf{x}_1 \in \mathcal{D}_t \, \mathbf{x}_2 \in \mathcal{D}_a \\
e(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2) &= \exp\{\eta_2 \cos(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2)\} \quad \forall \mathbf{x}_1 \in \mathcal{D}_a \, \mathbf{x}_2 \in \mathcal{D}_t,
\end{aligned}
\tag{1}
$$

where $\hat{\mathbf{x}} = \phi(\mathbf{x})$, $\hat{\mathbf{x}}_1 = \phi(\mathbf{x}_1)$, $\hat{\mathbf{x}}_2 = \phi(\mathbf{x}_2)$, η_1, η_2 are hyperparameters to increase the probability of finding source/target nodes depending on the direction of the random walk as introduced later, and $\cos(\cdot, \cdot)$ denotes the cosine similarity between two vectors, matrices or tensors of the same size with its definition as $\cos(\mathbf{z}_1, \mathbf{z}_2) = \langle \mathbf{z}_1, \mathbf{z}_2 \rangle / \sqrt{\langle \mathbf{z}_1, \mathbf{z}_1 \rangle \langle \mathbf{z}_2, \mathbf{z}_2 \rangle}$ where $\langle \cdot, \cdot \rangle$ denotes the dot product. Based on Eq. (1), we can

see that there is no self loop in this graph. Moreover, there is no edge between the source and target samples and this is because the two domains have a large discrepancy, leading to unreliable similarities. Note that during the optimization process, $\phi(\cdot)$ changes over epochs and so do the edge weights in the graph G.

Based on the graph G, the random walk works as follows. Suppose we want to traverse the nodes in a graph. At current node i, the probability to visit node j next is proportional to $e(i, j)$. So the random walk will start at a node and then randomly visit the next node with such probability until reach some goal node.

In the DERWENT model, we can construct the random walk in two directions. The random walk of the first type starts at a node corresponding to a data point in the source domain, and then randomly visits one of its neighbors with the probability proportional to edge weights. This process will continue until reaching a node in the target domain or the number of nodes visited exceeds a threshold denoted by θ. The random walk of the second type acts similarly but it will start at a node in the target domain and will stop when reaching a node in the source domain or the number of nodes visited exceeds θ.

We first introduce how to learn from the first type in the DERWENT model. Given a mini-batch which contains a subset of data points from the source, auxiliary, and target domains, we first construct a graph G on this mini-batch with η_1 being 1 and η_2 being η that is a hyperparameter. Here η is required to be larger than 1. Such setting will increase the probability to find a target instance in the neighborhood during the random walk and shorten the length of sequences in the random walk. Then we conduct random walk on the graph G to sample several sequences with different starting nodes. For the ith sequence $\mathcal{S}_i = (\hat{\mathbf{x}}_{i,1}, \ldots, \hat{\mathbf{x}}_{i,n_{s_i}})$ where n_{s_i} denotes the length of this sequence or equivalently the number of nodes visited satisfying $n_{s_i} \le \theta$, we expect two neighboring data points to be similar, which is to help learn the feature extraction network, and define the corresponding similarity loss as

$$
\begin{aligned}
l_{i,1} = \sum_{j=1}^{n_{s_i}-1} \Big(&- \ln \sigma_\alpha(\cos(\hat{\mathbf{x}}_{i,j}, \hat{\mathbf{x}}_{i,j+1})) \\
&- \ln(1 - \sigma_\alpha(\cos(\hat{\mathbf{x}}_{i,j}, \hat{\mathbf{z}}_{i,j}))) \Big),
\end{aligned}
\tag{2}
$$

where $\sigma_\alpha(x) = \frac{1}{1+\exp\{-\alpha x\}}$ denotes a scaled sigmoid function to make the output spread more over $[0, 1]$ due to the limited range (i.e., $[-1, 1]$) of the cosine similarity, and $\hat{\mathbf{z}}_{i,j}$ is sampled randomly out of $\mathcal{S}_i$ but in the mini-batch to act as a dissimilar data point to $\hat{\mathbf{x}}_{i,j}$.

For $\mathcal{S}_i$, if the last data point is from the target domain, which means that the random walk finds a path from the source domain to the target domain, we expect that this target data point can be represented based on other data points in this sequence since nodes in a sequence generated by deep random walk are inherently related based on the hidden feature representation. To achieve that, we use the sequence of $\mathcal{S}_i$ except $\hat{\mathbf{x}}_{i,n_{s_i}}$ to reconstruct it and hence we can formulate
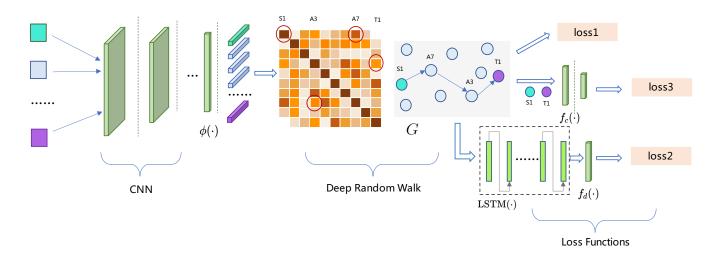
Figure 2: The architecture of the DERWENT model can be divided into three parts: (1) Images features are extracted by a pre-trained deep convolution neural network (CNN) followed by the feature extraction network $\phi(\cdot)$. (2) According to the hidden feature representation generated in the previous step, we construct a graph G on each mini-batch data points and adopt the random walk to generate transfer paths. (3) Three losses are calculated based on the sampled sequences.

the corresponding sequence loss as

$$l_{i,2} = \|\hat{\mathbf{x}}_{i,n_{s_i}} - f_d(\text{LSTM}(\hat{\mathbf{x}}_{i,1}, \ldots, \hat{\mathbf{x}}_{i,n_{s_i}-1}))\|, \quad (3)$$

where $\text{LSTM}(\cdot)$ denotes an LSTM to output the hidden state of the last position and $f_d(\cdot)$ is a neural decoder to generate an approximation of $\hat{\mathbf{x}}_{i,n_{s_i}}$. Here the LSTM is used since $\hat{\mathbf{x}}_{i,1}, \ldots, \hat{\mathbf{x}}_{i,n_{s_i}-1}$ form a sequence.

Moreover, by defining the set of labeled data in $\mathcal{S}_i$ from either source or target domain as $\mathcal{L}_i$, we formulate the classification loss as

$$l_{i,3} = \sum_{(\hat{\mathbf{x}},y) \in \mathcal{L}_i} w(\hat{\mathbf{x}}) \Big(-(1-y)\ln(1 - \sigma(f_c(\hat{\mathbf{x}})))$$
$$- y\ln(\sigma(f_c(\hat{\mathbf{x}}))) \Big),$$

where $\sigma(x) = \frac{1}{1+\exp\{-x\}}$ denotes the sigmoid function, $f_c(\cdot)$ denotes the classification network, and $w(\hat{\mathbf{x}})$, a measure of the instance importance, is equal to 1 for a target data point and otherwise a positive value smaller than 1. For a source data point $\hat{\mathbf{x}}$, we define $w(\hat{\mathbf{x}})$ as $w(\hat{\mathbf{x}}) = \sigma_\alpha(\cos(\hat{\mathbf{x}}, \hat{\mathbf{x}}_{i,n_{s_j}}))$, which reflects the confidence to use the loss of a source data point for the target domain.

By combining the above three losses, the objective function of the DERWENT model with the first type of the random walk is formulated as

$$\min \sum_i l_{i,1} + o_i(\lambda_1 l_{i,2} + \lambda_2 l_{i,3}), \quad (4)$$

where λ_1, λ_2 are regularization parameters, and the indicator o_i equals 1 when $\mathcal{S}_i$ reaches a target data point and otherwise 0. Hence the sequence loss and classification loss are only used when the sequence reaches a target data point. Parameters to be optimized in problem (4) include those in the feature extraction network $\phi(\cdot)$, the $\text{LSTM}(\cdot)$, the neural decoder $f_d(\cdot)$, and the classification network $f_c(\cdot)$.

The second type of the random walk can be formulated similarly with slight differences. To increase the probability to reach a source node in the graph G and shorten the length of sequences, η_1 and η_2 are set to η and 1, respectively, where η is used in the first type of the random walk. The similarity loss (i.e., $l_{i,1}$) has no change. The sequence loss (i.e., $l_{i,2}$) is formulated similarly with the ending data point $\hat{\mathbf{x}}_{i,n_{s_i}}$ from the source domain being represented by other data points in the same sequence. In the classification loss (i.e., $l_{i,3}$), the starting target data point will contribute the classification loss no matter whether the sequence reaches the source domain, which is different from the first type as the label information in the target domain is highly valued.

The entire objective function of the DERWENT model is to sum those of the two types up. The architecture of the DERWENT model is illustrated in Figure 2.

Discussion

Different from DDTL, the DERWENT model identifies the transfer path $\{\mathcal{S}_i\}$ between the source and target domains in two directions based on the random walk technique, learns good feature extraction network via the similarity and sequence losses (i.e., $l_{i,1}$ and $l_{i,2}$), and reuses the label information in the source domain by defining the weight function $w(\cdot)$ based on the learned feature extraction network. Different from the DeepWalk method which maximizes the co-occurrence probability among the data nodes, the DERWENT method maximizes the similarity between adjacent data points in a sampled sequence via the similarity loss and minimizes the reconstruction error of the ending data point via the sequence loss.

Experiments

In this section, we conduct experiments to evaluate the proposed DERWENT model.

Experimental Settings

We conduct experiments on three benchmark datasets, including the Animals with Attributes (AwA) dataset (Xian et al. 2019), the Caltech-256 dataset (Griffin, Holub, and Perona 2007), and the CIFAR-100 dataset (Krizhevsky and Hinton 2009). The AwA dataset contains 30,475 pictures with 50 categories, where the number of instances per class varies from 92 to 1,168. We select one of three categories including 'chihuahua', 'sheep' and 'lion' to form the positive class of the source domain, and select one of six categories including 'antelope', 'chimpanzee', 'rabbit', 'bobcat', 'pig' and 'german+shepherd' as the positive class of the target domains. Moreover, by following (Tan et al. 2017), we mix data from seven categories 'beaver', 'blue+whale', 'mole', 'mouse', 'ox', 'skunk', and 'weasel' to form the negative class for source and target domains but with no overlapping. Data of all the remaining categories are used as auxiliary domains. The Caltech-256 dataset contains 30,607 images with 257 categories, including a background category 'clutter'. There are 80 to 827 images in each category. To validate the performance between distant domains, we select some relatively different categories to form the source and target domains, such as 'baseball-bat', 'conch', 'airplane', 'skateboard', 'soccer-ball', 'horse' and 'gorilla'. Specifically, we first randomly select a category as the positive class of the source domain and then randomly select another category to be the positive class of the target domain. Data in the 'clutter' category are randomly selected to form negative instances for both source and target domains but with no overlapping. Data of all the remaining categories are used as auxiliary domains. The CIFAR-100 dataset contains 100 classes, where the number of instances per class is 500. We select one category from 'chair', 'bus', 'rose', 'woman', and 'bottle' to form the positive class of the source domain and select one of three categories including 'cup', 'phone' and 'bowl' as the positive class of the target domain.

We mix data from categories related to aquatic mammals including 'beaver', 'dolphin', 'otter', 'seal' and 'whale' to form negative examples for source and target domains with no overlapping. Data of all the remaining categories are used as auxiliary domains.

According to the above construction of different domains, on the AwA dataset, we have 9 distant transfer learning tasks, including 'chihuahua-to-bobcat' (C→B), 'chihuahua-to-antelope' (C→A), 'chihuahua-to-pig' (C→P), 'sheep-to-rabbit' (S→R), 'sheep-to-chimpanzee' (S→CH), 'sheep-to-german+shepherd' (S→SH), 'lion-to-rabbit' (L→R), 'lion-to-chimpanzee' (L→CH), and 'lion-to-german+shepherd' (L→SH). On Caltech-256 dataset, we have 6 distant transfer learning tasks, including 'airplane-to-soccer-ball' (A→S), 'gorilla-to-baseball-bat' (G→B), 'airplane-to-skateboard' (A→SK), 'horse-to-conch' (H→C), 'soccer-ball-to-skateboard' (S→SK), and 'soccer-ball-to-conch' (S→C). On the CIFAR-100 dataset, we have 5 distant transfer learning tasks, including 'bus-to-phone' (B→P), 'chair-to-cup' (C→CU), 'rose-to-phone' (R→P), 'bottle-to-bowl' (BT→BW), and 'woman-to-phone' (W→P).

Baseline models in comparison include a deep neural network (DNN) which is trained on the target data only, DAN (Long et al. 2015), DANN (Ganin et al. 2016), CNN-based STL (Kemker and Kanan 2017), and DDTL. We also compare with two varaints of the proposed DERWENT method denoted by *DERWENT w/o L1* and *DERWENT w/o LSTM* by discarding the similarity loss defined in Eq. (2) and the sequence loss defined in Eq. (3), respectively. We use the VGG-11 model (Simonyan and Zisserman 2015) pre-trained on the ImgeNet dataset as the backbone network followed by the feature extraction network $\phi(\cdot)$, which has a Fully-Connected (FC) layer with 256 hidden units and the activation function as the tanh function. We use the same network structure for all the baseline models. In the DERWENT model, a one-layer bi-directional LSTM with 128 hidden units is used to compute the sequence loss defined in Eq. (3), the neural decoder $f_d(\cdot)$ has a FC layer with 256 outputs, and the classification network $f_c(\cdot)$ has a FC layer with 2 outputs. For optimization, we use the mini-batch SGD with the Nestorov momentum 0.9. The batch size is set to 128, including 10, 8, and 110 in the source, target and auxiliary domains, respectively. The learning rate is set to 0.01. η in the graph (i.e., Eq. (1)) is initialized to 1.1 and then increased according to epochs as $1.1^{\lfloor \text{epochs}/3 \rfloor}$. All the regularization parameters in the DERWENT model are set to 1.

Experimental Results

In each experiment, we randomly selected 10 labeled instances in each class in the target domain for training and the rest for testing. Each setting is repeated for three times and the average results are reported in Tables 1-3. According to the results, we can see that in most cases the accuracies of DAN and DANN, which are transfer learning methods, are

Method	C→B	C→A	C→P	S→R	S→CH	S→SH	L→R	L→CH	L→SH	Avg
DNN	83.5	89.1	65.0	87.3	76.0	77.5	87.3	76.0	77.5	79.9
DAN	85.3	88.7	63.7	88.2	63.8	76.1	82.3	64.5	80.3	77.0
DANN	82.9	86.2	61.9	78.5	68.2	73.4	71.1	72.1	76.7	74.6
STL	83.1	89.8	64.4	89.1	79.9	80.5	89.1	79.9	80.5	81.8
DDTL	85.6	92.9	77.2	89.3	72.5	79.2	91.8	78.7	80.8	83.1
DERWENT w/o L1	90.0	93.4	74.3	92.6	89.0	90.9	93.6	87.7	90.0	89.1
DERWENT w/o LSTM	**91.3**	96.1	75.9	94.2	85.2	**91.9**	93.2	88.7	**92.7**	89.9
DERWENT	90.3	**96.3**	**77.9**	**94.6**	**92.7**	91.8	**93.8**	**89.4**	92.0	**91.0**

Table 1: Accuracy (%) of various models on different tasks of the AwA dataset.

Method	A→S	G→B	A→SK	H→C	S→SK	S→C	Avg
DNN	82.9	72.6	66.7	82.8	66.7	82.8	75.8
DAN	85.4	68.4	76.4	84.5	63.4	83.3	76.9
DANN	79.3	70.9	65.6	84.9	61.3	82.8	74.1
STL	84.1	76.1	69.9	75.3	69.9	75.3	75.1
DDTL	84.1	71.8	78.5	89.2	61.3	84.9	78.3
DERWENT w/o L1	89.0	82.1	83.9	89.2	74.2	88.2	84.4
DERWENT w/o LSTM	**90.8**	80.3	81.7	**90.3**	**77.4**	89.2	85.0
DERWENT	**90.8**	**85.4**	**84.9**	87.1	**77.4**	**91.4**	**86.2**

Table 2: Accuracy (%) of various models on different tasks of the Caltech-256 dataset.

Method	B→P	C→CU	R→P	BT→BW	W→P	Avg
DNN	89.7	87.4	89.7	87.2	89.7	88.7
DAN	89.3	79.6	81.9	85.8	93.4	86.0
DANN	90.1	73.0	74.2	76.7	91.3	81.1
STL	89.2	87.8	89.2	86.2	89.2	88.3
DDTL	91.5	86.0	88.0	83.3	94.4	88.6
DERWENT w/o L1	93.0	88.2	90.5	91.5	95.1	91.7
DERWENT w/o LSTM	93.4	**91.3**	92.2	90.5	96.3	92.7
DERWENT	**93.8**	91.1	**93.0**	**91.5**	**96.9**	**93.3**

Table 3: Accuracy (%) of various models on different tasks of the CIFAR-100 dataset.
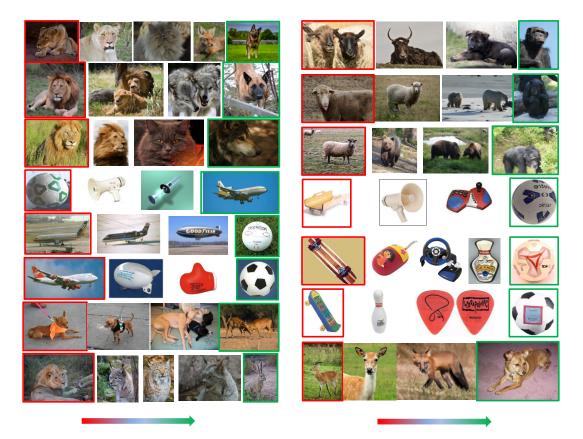

Figure 3: Some transfer paths generated by the DERWENT method. Specifically, each row in both columns represents a transfer path from the source domain in a red rectangle to the target domain in a green rectangle.
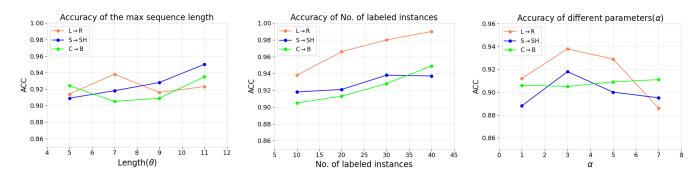
Figure 4: Sensitivity analysis of hyperparameters of the DERWENT model.

lower than that of DNN. It is because that there is a large discrepancy between the source and target domains which results in 'negative transfer' for the traditional transfer learning methods. The STL method performs slightly better than DNN as STL can learn a useful feature representation from auxiliary domains. As a distant transfer learning method, DDTL performs better than DNN, DAN, DANN, and STL as it uses auxiliary domains as a bridge to help transfer the knowledge contained in the source domain to help the learning in the target domain. The variants of the proposed DERWENT method perform worse than DERWENT method but better than others, implying that each loss function defined in the DERWENT method is necessary. Among all the methods in comparison, the proposed DERWENT method performs the best, which demonstrates the effectiveness of the proposed DERWENT method.

Visualization of Transfer Paths

In order to understand how the proposed DERWENT method transfers knowledge between distant domains through auxiliary domains, we visualize the transfer paths obtained by the DERWENT method in Figure 3. According to Figure 3, we can see that in each path, the source image in a red rectangle is completely different from the target image in a green rectangle, and from left to right, images visited by the random walk are gradually close to the target image. For example, in two tasks 'airplane-to-soccer-ball' (located in the third row from the bottom in the left column of Figure 3) and 'skateboard-to-soccer-ball' (located in the second row from the bottom in the right column of Figure 3), the source and target domains are significantly different. The DERWENT method first relates source images to the 'blimp' and 'bowling-pin' classes, respectively, which are similar to source images, and then gradually visits images that are more similar to the target domain until reaching some target images. Hence, based on transfer paths, we can understand how the DERWENT model works and this can improve the model interpretability.

Sensitivity Analysis

To test the sensitivity of the performance of the DERWENT model with respect to different hyperparameters including the maximum length θ of sampled sequences in the random walk, the number of labeled instances in each class of the

target domain, and α used in the similarity loss (i.e., Eq. (2)), we conduct experiments for each hyperparameter by fixing other hyperparameters on three distant transfer learning tasks, including L→R, S→SH, and C→B. According to the results shown in Figure 4, we can see that θ has little effect on the performance and one possible reason is that the random walk has reached the destination in less than θ steps. It is obvious that more labeled instances in the target domain lead to better performance. Moreover, according to the results, setting α to 3 has the best performance for the three tasks and this is the setting for α in all the experiments.

Conclusion

To solve the distant transfer learning problem, we propose the DERWENT method based on deep random walk, which can help transfer knowledge from the source domain to the target domain across auxiliary domains gradually. Different from existing methods, the proposed DERWENT method can automatically find the transfer paths. The proposed DERWENT method has shown state-of-the-art performance in three benchmark image datasets.

In the future research, we are interested in extending the DERWENT model to deal with multiple source domains.

Acknowledgements

References

Dai, W.; Yang, Q.; Xue, G.; and Yu, Y. 2007. Boosting for transfer learning. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, 193–200.

Gan, J.; Li, L.; Zhai, Y.; and Liu, Y. 2014. Deep self-taught learning for facial beauty prediction. *Neurocomputing* 144: 295–303.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. S. 2016. Domain-Adversarial Training of Neural Networks. *J. Mach. Learn. Res.* 17: 59:1–59:35.

Griffin, G.; Holub, A.; and Perona, P. 2007. Caltech-256 Object Category Dataset. Technical Report CNS-TR-2007-001, California Institute of Technology.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation* 9(8): 1735–1780.

Hu, D. H.; and Yang, Q. 2011. Transfer Learning for Activity Recognition via Sensor Mapping. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 1962–1967.

Kemker, R.; and Kanan, C. 2017. Self-Taught Feature Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote. Sens.* 55(5): 2693–2705.

Khan, M. N. A.; and Heisterkamp, D. R. 2016. Adapting instance weights for unsupervised domain adaptation using quadratic mutual information and subspace learning. In *23rd International Conference on Pattern Recognition*, 1560–1565.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases* 1(4).

Kumar, M. P.; Packer, B.; and Koller, D. 2010. Self-Paced Learning for Latent Variable Models. In *Advances in Neural Information Processing Systems 23*, 1189–1197.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. I. 2015. Learning Transferable Features with Deep Adaptation Networks. In *Proceedings of the 32nd International Conference on Machine Learning*, 97–105.

Long, M.; Wang, J.; Cao, Y.; Sun, J.; and Yu, P. S. 2016. Deep Learning of Transferable Representation for Scalable Domain Adaptation. *IEEE Trans. Knowl. Data Eng.* 28(8): 2027–2040.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space. In *Workshop Track Proceedings of the 1st International Conference on Learning Representations*.

Ng, M. K.; Wu, Q.; and Ye, Y. 2012. Co-transfer learning via joint transition probability graph based method. In *In Proceedings of the 1st International Workshop on Cross Domain Knowledge Discovery in Web and Social Network Mining*, 1–9.

Pan, S. J.; Shen, D.; Yang, Q.; and Kwok, J. T. 2008. Transferring Localization Models across Space. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, 1383–1388.

Pan, S. J.; Tsang, I. W.; Kwok, J. T.; and Yang, Q. 2009. Domain Adaptation via Transfer Component Analysis. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 1187–1192.

Pan, S. J.; and Yang, Q. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22(10): 1345–1359.

Pan, W.; Liu, N. N.; Xiang, E. W.; and Yang, Q. 2011. Transfer Learning to Predict Missing Ratings via Heterogeneous User Feedbacks. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 2318–2323.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. DeepWalk: Online learning of social representations. In *Proceedings of*

The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 701–710.

Raina, R.; Battle, A.; Lee, H.; Packer, B.; and Ng, A. Y. 2007. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, 759–766.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proceedings of the 3rd International Conference on Learning Representations*.

Tan, B.; Song, Y.; Zhong, E.; and Yang, Q. 2015. Transitive Transfer Learning. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1155–1164.

Tan, B.; Zhang, Y.; Pan, S. J.; and Yang, Q. 2017. Distant Domain Transfer Learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2604–2610.

Tan, B.; Zhong, E.; Ng, M. K.; and Yang, Q. 2014. Mixed-Transfer: Transfer Learning over Mixed Graphs. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, 208–216.

Tzeng, E.; Hoffman, J.; Darrell, T.; and Saenko, K. 2015. Simultaneous Deep Transfer Across Domains and Tasks. In *Proceedings of IEEE International Conference on Computer Vision*, 4068–4076.

Uribe, D. 2010. Domain Adaptation in Sentiment Classification. In *The Ninth International Conference on Machine Learning and Applications*, 857–860.

Xian, Y.; Lampert, C. H.; Schiele, B.; and Akata, Z. 2019. Zero-Shot Learning - A Comprehensive Evaluation of the Good, the Bad and the Ugly. *IEEE Trans. Pattern Anal. Mach. Intell.* 41(9): 2251–2265.

Yang, Q.; Zhang, Y.; Dai, W.; and Pan, S. J. 2020. *Transfer Learning*. Cambridge University Press.

Zhang, Y.; Liu, T.; Long, M.; and Jordan, M. I. 2019. Bridging Theory and Algorithm for Domain Adaptation. In *Proceedings of the 36th International Conference on Machine Learning*, 7404–7413.