Multi-Source Domain Adaptation with Distribution Fusion and Relationship Extraction

1st Keqiuyin Li  
Faculty of Engineering and IT  
University of Technology Sydney  
Sydney, NSW, Australia  
keqiuyin.li@student.uts.edu.au

2nd Jie Lu  
Faculty of Engineering and IT  
University of Technology Sydney  
Sydney, NSW, Australia  
jie.lu@uts.edu.au

3rd Hua Zuo  
Faculty of Engineering and IT  
University of Technology Sydney  
Sydney, NSW, Australia  
hua.zuo@uts.edu.au

4th Guangquan Zhang  
Faculty of Engineering and IT  
University of Technology Sydney  
Sydney, NSW, Australia  
guangquan.zhang@uts.edu.au

Abstract—Transfer learning is gaining increasing attention due to its ability to leverage previously acquired knowledge to assist in completing a prediction task in a similar domain. While many existing transfer learning methods deal with single source and single target problem without considering the fact that a target domain maybe similar to multiple source domains, this work proposes a multi-source domain adaptation method based on a deep neural network. Our method contains common feature extraction, specific predictor learning and target predictor estimation. Common feature extraction explores the relationship between source domains and target domain by distribution fusion and guarantees the strength of similar source domains during training, something which has not been well considered in existing works. Specific predictor learning trains source tasks with cross-domain distribution constraint and cross-domain predictor constraint to enhance the performance of single source. Target predictor estimation employs relationship extraction and selective strategy to improve the performance of the target task and to avoid negative transfer. Experiments on real-world visual datasets show the performance of the proposed method is superior to other deep learning baselines.

I. INTRODUCTION

Transfer learning [1], [2] has been explored for many years and gained success in a variety of applications in real-world scenarios, such as nature language processing, computer vision and biological problems. The main goal of transfer learning is to improve the performance of the target task using the knowledge learned from a similar source domain, since the target training data is often difficult to collect or expensive to label, especially where medicine is involved. Employing different transfer information, it can be divided into four categories: instance-based method [3], feature-based method [4], parameter-based method [5] and relationship-based method [6].

Based on feature and parameter transformation, one popular technique to achieve transfer learning is domain adaptation [7]–[9], which aims to tackle domain shift by reducing the discrepancy of distributions between the source and target domains in a latent feature space, including adaptations of marginal distribution [10], conditional distribution [11] and joint distribution [12].

Zellinger et al. [13] proposed a novel metric function named central moment discrepancy to measure the distance between probability distributions which can be solved without highly complex and costly kernel matrix computations. Zuo et al. [14], [15] used fuzzy system and granular computing to achieve regression transfer in homogeneous and heterogeneous feature spaces. Benefiting from the development of deep learning, recent surveys employed deep neural networks to extract common features of source and target domains. Based on adversarial learning, Long et al. [16] proposed joint adaptation networks which used joint maximum mean discrepancy to align the joint distributions across domains in hidden specific layers. Liu et al. [17] and Jang et al. [18] explored the transferability of deep feature representations and provided experiments and theory analysis on what and where to transfer in deep networks.

However, all these studies focused on single source domain adaptation while, in practice, a target domain can be similar to multiple source domains, and they may be different from each other but could provide richer information for transfer. It is easy to see why multi-source domain adaptation now attracts greater attention. Zhao et al. [19] and Wen et al. [20] measured the discrepancy by \( H \)-divergence and proposed an adversarial strategy based deep framework to solve multiple sources domain adaptation both for classification and regression problems. Guo et al. [21] proposed a mixture-of-experts method to do text classification and speech tagging with multiple sources, where Mahalanobia distance and confidence score were used to extract the relationship between each source domain and target domain. Ding et al. [22] tried to achieve domain adaptation with multiple incomplete source domains via low-rank matrix which could recover missing labels in source domains based on latent features from a target
domain. Redko et al. [23] and Xu et al. [24] solved multiple sources domain adaptation under target shift and category shift, where source labels might not completely share labels of target domain or share labels with different proportions. Zhu et al. [25] proposed a two-stage alignment framework for multiple sources domain adaptation, in which domain-specific distribution alignment was used to reduce discrepancy between source domains and target domain; domain-specific classifier alignment was used to reduce difference among all classifiers.

The main idea of all described multi-source domain adaptation begins with extracting common features of source domains and target domain before training specific predictors of each source domain, finally combining all specific predictors as the target predictor. The popular combination rules include the average of source predictors and weighted average of source predictors. Although some of these approaches have considered the connections between different source domains and target domain, they still treat each source domain equally during training. It has been proven that adding irrelevant source samples may lead to negative transfer and reduce the performance of the learning task [26]. Measuring relationship among source domains and target domain is necessary to avoid negative performance. In this paper, in order to learn weights of multiple sources to get high performance of predictor in target domain, following work [25] and domain matching method proposed by Li et al. [27], we proposed a deep neural network based multi-source domain adaptation approach with distribution fusion and relationship extraction. This measures relatedness among sources and target during training and guarantees that more similar sources contribute more to target task learning. Our main contributions are as follows:

- We propose a method using distribution fusion and relationship extraction to guarantee the strength of similar source domains during learning a target task, something which is not well considered in existing methods;
- We use a selective strategy to choose the best performance of the target task and to avoid negative transfer of irrelevant sources.

The structure of this paper is designed as follows: Section II describes the proposed model. Section III carries a series of experiments on real-world datasets and analyses the results. Our conclusion is given in section IV.

II. PROPOSED METHOD

The proposed method contains the following three parts: (1) common feature extraction where pre-trained deep neural network, fine-tuning operation and distribution fusion strategy are employed to collect robust features; (2) specific predictor learning where specific domain adaptation and predictor of each source domain are learned with cross-domain constraints; (3) target predictor estimation where relationship extraction and selective strategy are used to obtain target predictor without negative transfer. The whole procedure is shown in Fig. 1.

A. Common Feature Extraction

In general, domain adaptation is an unsupervised learning task. For single source domain adaptation, given source domain $D_s = \{(x^s_i, y^s_i)\}_{i=1}^{n_s}$ and target domain $D_t = \{x^t_i\}_{i=1}^{n_t}$, where $x^s, x^t \in X$ represent samples, $y^s \in \mathcal{Y}$ indicates corresponding label of $x^s$ and $n_s, n_t$ indicate the number of samples in source domain and target domain respectively. The main step is mapping distributions of source domain and target domain. One popular method to achieve this is maximum mean discrepancy (MMD) [10] which can be formulated as:

$$MMD(X_s, X_t) = \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(x^s_i) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(x^t_j)$$

where $\|F\|_H$ is reproducing kernel Hilbert space (RKHS) norm, $\phi$ is kernel-induced feature map.

In our setting, when there are $K$ source domains $\{D_{s_k}\}_{k=1}^{K}$ to map original samples into a common feature space where domain adaptation can be achieved, we use a pre-trained deep neural network $F_s$ to collect latent features of all domains. At the same time, fine-tuning block $F_{fc}$, which is optimized by distribution fusion block $F_{d}$ and specific domain adaptation block (we will explain it in detail in section II-B), is added to fine-tune convolution layers in order to extract more robust latent representations. The extracted latent features are then used to complete specific domain adaptation. Common feature extraction can be formulated as:

$$f^i_{c_{sk}} = F_{fc}(F_{s}(x^s_{ki})), f^j_{c_{tk}} = F_{fc}(F_{s}(x^t_{ji})), i = 1, 2, \ldots, n_{sk}, j = 1, 2, \ldots, n_{tk}$$

Rewrite (1) according to the proposed common feature extractor and express it as multi-source setting, that is:

$$MMD(X_{s_k}, X_t) = \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(f^i_{c_{sk}}) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(f^j_{c_{tk}})$$

Before fine-tuning common features extracted by (2) using specific domain adaptation block and estimating RKHS distance between each source domain and target domain, we first add the distribution fusion block to optimize the parameters of the common feature fine-tuning block. So, corresponding loss function can be written as:

$$L_{F_d} = MMD(\hat{X}_{sk}, X_t)$$

$$= \frac{1}{n_{sk}} \sum_{i=1}^{n_{sk}} \phi(f^i_{c_{sk}}) - \frac{1}{n_{sk}} \sum_{j=1}^{n_{sk}} \phi(f^j_{c_{sk}})$$

$$= \sum_{k=1}^{K} \omega_k \left( \frac{1}{n_{sk}} \sum_{i=1}^{n_{sk}} \phi(f^i_{c_{sk}}) - \frac{1}{n_{sk}} \sum_{j=1}^{n_{sk}} \phi(f^j_{c_{sk}}) \right)$$

where $k$ is the $k$th source domain, $\hat{X}_s$ means weighted combination of source domains, which contributes the update
of parameters of common feature fine-tuning block and gives more similar sources larger weights for subsequent training.

B. Specific Predictor Learning

After extracting latent features of all domains, which represent common knowledge of the sources and the target, specific domain adaptation blocks are used to learn specific features and predictors of each source domain simultaneously. The specific domain adaptation block of each source domain contains three fine-tuning layers and a fully connected layer. The specific fine-tuning layers \( \{ F_{fs}^k \}_{k=1}^K \) are added to learn specific latent features of corresponding \( k \)th source domain and the target domain, which guarantee that the predictor trained on the source domain can be used on the target domain. The fully connected layer is employed to learn high performance predictors of each source domain with cross-domain constraints. The final specific features of each source domain are:

\[
\begin{align*}
    f_{s_{sk}}^i & = F_{fs}^k(f_{csk}^i), \\
    f_{s_{tk}}^j & = F_{fs}^k(f_{ct}^j),
\end{align*}
\]

(5)

\( f_{s_{tk}} \) means feeding common representation \( f_{cs} \) into the specific feature extractor of \( k \)th source domain.

To optimize \( F_{fs} \), here we use the same strategy proposed in [25], cross-domain constraints contain two parts: cross-domain distribution and cross-domain predictor. It can be expressed as:

\[
\begin{align*}
    L_{dc} & = \frac{1}{K} \sum_{k=1}^{K} \text{MMD}(X_{sk}, X_t) = \\
    & = \frac{1}{K} \sum_{k=1}^{K} \left\| \frac{1}{n_{sk}} \sum_{i=1}^{n_{sk}} \phi(f_{s_{sk}}^i) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(f_{s_{tk}}^j) \right\|_H^2,
\end{align*}
\]

(6)

and

\[
\begin{align*}
    L_{cp} & = \frac{2}{K(K-1)} \sum_{k_1=1}^{K-1} \sum_{k_2=k_1+1}^{K} \\
    & \left( \frac{1}{n_t} \sum_{j=1}^{n_t} \left| F_{pres}^k(f_{s_{sk_1}}^j) - F_{pres}^k(f_{s_{sk_2}}^j) \right| \right),
\end{align*}
\]

(7)

where \( F_{pres} \) represents a specific predictor of the source domain. By minimizing (6) and (7), specific domain-invariant features can be learned by taking generated cross-domain distributions into consideration. In addition, target samples close to class edges are more probably to get the same prediction results. It has been proven that adding these cited cross-domain constrains brings positive effects [25].
The final specific predictor $F_{\text{pres}}^k$ of each source domain is controlled by cross entropy, that is:

$$L_{\text{pres}} = \sum_{k=1}^{K} \sum_{i=1}^{n_k} y_{sk}^{i} \log (F_{\text{pres}}^k(f_{sk}^i)), \quad (8)$$

The total loss function of proposed model is:

$$L = L_{\text{pres}} + \lambda L_{F_\theta} + \gamma L_{dc} + \mu L_{cp}. \quad (9)$$

C. Target Predictor Estimation

After obtaining specific predictors of source domains, it is important to choose an appropriate combination rule to estimate the prediction function of the target domain. Since we have learned a relationship between latent features of target domain and the combination of weighted source domains using distribution fusion block, here we use the learned weights to extract the connection among source predictors and target predictor. To further strengthen the performance of outputs, the predictive accuracy of single source domain adaptation is taken into account. The final target predictor is:

$$F_{\text{pret}} = \sum_{k=1}^{K} \alpha_k F_{\text{pres}}^k, \quad (10)$$

where

$$\alpha_k = \frac{\text{sigmoid}(\omega_k) \cdot w_k}{\sum_{k'=1}^{K} \text{sigmoid}(\omega_{k'}) \cdot w_{k'}},$$

$$w_k = \frac{\sum_{k'=1}^{K} \text{accu}_k'}{\text{accu}_k}.$$  

$\omega_k$ is the parameter of distribution fusion block which indicates the similarity between $k$th source domain and target domain and $\text{accu}_k$ means the prediction accuracy of $k$th source domain predictor.

To avoid negative transfer, source selective strategy is employed to obtain a high performance of final outputs of the target domain. Our proposed method compares single source outputs directly with predictions of target predictor (10). If there is negative transfer, the final target predictor equals to the single source predictor which achieves the best performance.

The whole method is summarized as follows.

III. EXPERIMENTS

A. Datasets and Setting

To evaluate the efficiency of the proposed multi-source domain adaptation method, we take a series of experiments with some existing state-of-the-art baselines on datasets ImageCLEF-DA and Office-31.

ImageCLEF-DA is a balanced dataset which contains 1800 images from 12 categories shared by datasets Caltech-256 (C), ImageNet ILSVRC 2012 (I) and Pascal VOC 2012 (P) and each dataset is regarded as a domain. Every category contains 50 images and there are 600 images in each domain. We test the proposed model by building three tasks: $I, C \rightarrow P$; $I, P \rightarrow C$; $C, P \rightarrow I$.

Office-31 is an unbalanced dataset which comprises 4110 images from 31 categories shared by datasets Amazon (A), Webcam (W) and DSLR (D) and, again, each dataset is regarded as a domain. Amazon contains 2817 images from amazon.com, Webcam has 795 images taken by web camera and DSLR holds 498 images taken by digital SLR camera. The number of images in each category is different. We test the proposed model by building three tasks: $A, W \rightarrow D$; $A, D \rightarrow W$; $D, W \rightarrow A$.

Since there are few methods dealing with multi-source domain adaptation on real-world visual recognition, we compare the proposed method with the two most recent multiple source domain adaptation methods: Deep Cocktail Network (DCTN) [24] and Multiple Feature Spaces Adaptation Network (MF-SAN) [25], along with other efficient single domain adaptation methods: ResNet [28], Deep Adaptation Network (DAN) [29], Deep Coral (D-CORAL) [30] and Reverse Gradient (RevGrad) [31]. For these single source domain adaptation models, we use two standards which are the same as used in previous surveys: simply combining all source domains as one domain, that is “Source Combine”; returning the best single source domain adaptation results, that is “Single Best”. For multi-source domain adaptation models, the standard is labelled “Multi-Source”, which indicates transfer results of multiple sources.

We choose ResNet50 as a pre-train model; common feature fine-tuning block contains 3 convolution layers, with kernel sizes of $1 \times 1, 3 \times 3, 1 \times 1$; distribution fusion block contains 1 fully connected layer; specific feature fine-tuning block includes 3 convolution layers, with kernel sizes of $1 \times 1, 3 \times 3,
specific predictor comprises 1 fully connected layer. The optimization algorithm is stochastic gradient descent (SGD), training batch size is 32; learning rate is 0.01; momentum is 0.9 and weight decay is $5e^{-4}$. Trade-off parameters $\lambda$, $\gamma$ and $\mu$ are set as the common value 0.5.

B. Comparison and Analysis

Since all baselines reported in previous survey are present only in the best performance, we also choose the best transfer results of the proposed method for evaluation. Tables I and II show the results of the proposed method trained with and without distribution fusion or common feature fine-tuning on the two datasets.

### Table I

Comparison of classification accuracy (%) of the proposed method with ResNet, DAN, D-CORAL, RevGrad, DCTN and MFSAN on ImageCLEF-DA dataset.

<table>
<thead>
<tr>
<th>Standards</th>
<th>Method</th>
<th>I, C-P</th>
<th>I, P-C</th>
<th>P, C-I</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>ResNet</td>
<td>74.8</td>
<td>91.5</td>
<td>83.9</td>
<td>83.4</td>
</tr>
<tr>
<td>Best</td>
<td>DAN</td>
<td>75.0</td>
<td>93.3</td>
<td>86.2</td>
<td>84.8</td>
</tr>
<tr>
<td></td>
<td>D-CORAL</td>
<td>76.9</td>
<td>93.6</td>
<td>88.5</td>
<td>86.3</td>
</tr>
<tr>
<td></td>
<td>RevGrad</td>
<td>75.0</td>
<td>96.2</td>
<td>87.0</td>
<td>86.1</td>
</tr>
<tr>
<td>Source</td>
<td>DAN</td>
<td>77.6</td>
<td>93.3</td>
<td>92.2</td>
<td>87.7</td>
</tr>
<tr>
<td>Combine</td>
<td>D-CORAL</td>
<td>77.1</td>
<td>93.6</td>
<td>91.7</td>
<td>87.5</td>
</tr>
<tr>
<td></td>
<td>RevGrad</td>
<td>79.9</td>
<td>93.7</td>
<td>91.8</td>
<td>87.8</td>
</tr>
<tr>
<td>Multi-Source</td>
<td>DCTN</td>
<td>75.0</td>
<td>95.7</td>
<td>90.3</td>
<td>87.0</td>
</tr>
<tr>
<td></td>
<td>MFSAN</td>
<td>79.1</td>
<td>95.4</td>
<td>93.6</td>
<td>89.4</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>79.5</td>
<td>95.8</td>
<td>93.7</td>
<td>89.7</td>
</tr>
</tbody>
</table>

### Table II

Comparison of classification accuracy (%) of the proposed method with ResNet, DAN, D-CORAL, RevGrad, DCTN and MFSAN on Office-31 dataset.

<table>
<thead>
<tr>
<th>Standards</th>
<th>Method</th>
<th>A, W-D</th>
<th>A, D-W</th>
<th>W, D-A</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>ResNet</td>
<td>99.3</td>
<td>96.7</td>
<td>62.5</td>
<td>86.2</td>
</tr>
<tr>
<td>Best</td>
<td>DAN</td>
<td>99.5</td>
<td>96.8</td>
<td>66.7</td>
<td>87.7</td>
</tr>
<tr>
<td></td>
<td>D-CORAL</td>
<td>99.7</td>
<td>98.0</td>
<td>65.3</td>
<td>87.7</td>
</tr>
<tr>
<td></td>
<td>RevGrad</td>
<td>99.1</td>
<td>96.9</td>
<td>68.2</td>
<td>88.1</td>
</tr>
<tr>
<td>Source</td>
<td>DAN</td>
<td>99.6</td>
<td>97.8</td>
<td>67.6</td>
<td>88.3</td>
</tr>
<tr>
<td>Combine</td>
<td>D-CORAL</td>
<td>99.3</td>
<td>98.0</td>
<td>67.1</td>
<td>88.1</td>
</tr>
<tr>
<td></td>
<td>RevGrad</td>
<td>99.7</td>
<td>98.1</td>
<td>67.6</td>
<td>88.5</td>
</tr>
<tr>
<td>Multi-Source</td>
<td>DCTN</td>
<td>99.3</td>
<td>98.2</td>
<td>64.2</td>
<td>87.2</td>
</tr>
<tr>
<td></td>
<td>MFSAN</td>
<td>99.5</td>
<td>98.5</td>
<td>72.7</td>
<td>90.2</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>99.6</td>
<td>98.7</td>
<td>73.1</td>
<td>90.5</td>
</tr>
</tbody>
</table>

It can be seen that the Source Combine results are better than Single Best results generally, getting a higher average accuracy than single source domain. Although the Single Best results of transfer tasks $I,P \rightarrow C$ using RevGrad and $A,W \rightarrow D$ using D-CORAL outperform other methods and standards, enriching data still has a positive influence on transfer performance in most cases. Multi-Source results commonly achieve greater degree than Source Combine except for DCTN. That may result from different problem settings, where DCTN is directly aimed at dealing with domain adaptation with multiple sources and category shift. However, in most multi-source domain adaptation surveys, there is no category shift in source domains. The proposed method obtains the highest classification accuracy than most baselines and standards. This indicates that exploring any relationship between source domains and target domain is important, and guaranteeing the strength of more similar sources is worthy undertaking to achieve the desired performance.

### Table III

Comparison of classification accuracy (%) of proposed method on dataset ImageCLEF-DA: M1: trained with common feature fine-tuning and distribution fusion; M2: trained with distribution fusion but without common feature fine-tuning; M3: trained with common feature fine-tuning and distribution fusion.

<table>
<thead>
<tr>
<th>Standards</th>
<th>Method</th>
<th>C-P</th>
<th>P-C</th>
<th>C-I</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>M1</td>
<td>77.5</td>
<td>92.2</td>
<td>93.3</td>
<td>87.7</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>79.5</td>
<td>96.2</td>
<td>94.0</td>
<td>89.9</td>
</tr>
<tr>
<td></td>
<td>M3</td>
<td>79.5</td>
<td>95.8</td>
<td>93.7</td>
<td>89.7</td>
</tr>
</tbody>
</table>

### Table IV

Comparison of classification accuracy (%) of proposed method on dataset Office-31: M1: trained with common feature fine-tuning but without distribution fusion; M2: trained with distribution fusion but without common feature fine-tuning; M3: trained with common feature fine-tuning and distribution fusion.

<table>
<thead>
<tr>
<th>Standards</th>
<th>Method</th>
<th>D-W</th>
<th>D-W</th>
<th>W, D-A</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>M1</td>
<td>99.6</td>
<td>97.9</td>
<td>72.0</td>
<td>89.8</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>100.0</td>
<td>98.6</td>
<td>72.7</td>
<td>90.4</td>
</tr>
<tr>
<td></td>
<td>M3</td>
<td>99.6</td>
<td>98.7</td>
<td>73.1</td>
<td>90.5</td>
</tr>
</tbody>
</table>

Tables III and IV test the effects of common feature fine-tuning and distribution fusion. It can be seen that training the proposed method with distribution fusion always achieves better performance than training the method with common feature fine-tuning only. For dataset ImageCLEF-DA which contains a small number of samples, adding depth of neural network may harm the performance. In contrast, for large scale dataset like Office-31, adding fine-tuning operation with distribution fusion can improve the performance.

IV. CONCLUSION AND FUTURE STUDY

In this paper, we proposed a domain adaptation method with multiple sources based on a pre-trained deep neural network. Our method uses fine-tuning operation and distribution fusion to explore the relationship between source domains and target domain while simultaneously guaranteeing that more similar sources contribute correspondingly more during training. Then the cross-domain distribution constraint and cross-domain predictor constraint are used to learn specific predictors of source domains. Target predictor is estimated by combining source predictors using relationship extraction. In conclusion, the selective strategy is used to avoid negative transfer. A series of experiments are designed and implemented on two real-world datasets. For more vigorous examination, the effects of
distribution fusion and fine-tuning are also tested. The results demonstrate the efficiency of our proposed.

The proposed method can still be improved and there are remaining problems that should be solved in future study. Initially, we currently use a fully connected layer to extract relationships between source domains and target domain, which requires significant memory space in a computer. Replacing it with convolution layers may improve calculation efficiency. This still needs to be explored in future research.

REFERENCES


