Unsupervised Domain Adaptation via Structured Prediction Based Selective Pseudo-Labeling

Qian Wang,[1] **Toby P. Breckon**[1,2]
[1]Department of Computer Science, Durham University, United Kingdom
[2]Department of Engineering, Durham University, United Kingdom
qian.wang173@hotmail.com, toby.breckon@durham.ac.uk

Abstract

Unsupervised domain adaptation aims to address the problem of classifying unlabeled samples from the target domain whilst labeled samples are only available from the source domain and the data distributions are different in these two domains. As a result, classifiers trained from labeled samples in the source domain suffer from significant performance drop when directly applied to the samples from the target domain. To address this issue, different approaches have been proposed to learn domain-invariant features or domain-specific classifiers. In either case, the lack of labeled samples in the target domain can be an issue which is usually overcome by pseudo-labeling. Inaccurate pseudo-labeling, however, could result in catastrophic error accumulation during learning. In this paper, we propose a novel selective pseudo-labeling strategy based on structured prediction. The idea of structured prediction is inspired by the fact that samples in the target domain are well clustered within the deep feature space so that unsupervised clustering analysis can be used to facilitate accurate pseudo-labeling. Experimental results on four datasets (i.e. Office-Caltech, Office31, ImageCLEF-DA and Office-Home) validate our approach outperforms contemporary state-of-the-art methods.

Introduction

Domain adaptation problems exist in many real-world applications. Unsupervised Domain Adaptation (UDA) aims to address problems where the unlabeled test samples and the labeled training samples are from different domains (dubbed target and source domains, respectively). Models learned with labeled samples from the source domain suffer from significant performance reduction when they are directly applied to the target samples without domain adaptation. Such reduced performance is mainly due to the domain shift characterized by the difference of data distributions between the two domains. To address this issue, approaches to UDA have been proposed trying to align the source and target domains by learning a joint subspace so that samples from either domain can be projected into this common subspace. Different algorithms were employed to promote the separability of target samples in such subspaces. The use of visual features extracted by deep models pre-trained on the large-scale ImageNet dataset (Deng et al. 2009) further facilitates these feature transformation based approaches. On the other hand, deep learning models are used to learn domain invariant features in an end-to-end manner. The gradient reversal layer (Ganin and Lempitsky 2015) and adversarial learning (Tzeng et al. 2017) have been employed for this purpose.

In the UDA problem, both labeled samples from the source domain and unlabeled samples from the target domain are assumed to be available during model training and hence it is a transductive learning problem. The opportunity of transductive learning is when the algorithm can explore the test data (samples from the target domain in the case of UDA). One effective method is to assign pseudo-labels to the target samples so that samples from both domains can be combined and ready for supervised learning. The accuracy of pseudo-labeling plays an important role in the whole learning process. In most existing methods, the target samples are pseudo-labeled independently once the classifier is trained overlooking the structural information underlying the target domain.

In this paper, we explore such structural information via unsupervised learning (i.e. K-means) and propose a novel UDA approach based on selective pseudo-labeling and structured prediction. Specifically, our approach tries to learn a domain invariant subspace by Supervised Locality Preserving Projection (SLPP) using both labeled source data and pseudo-labeled target data. The accuracy of pseudo-labeling is promoted by structured prediction and progressively selections. The contributions of this work can be summarized as follows:

– a novel iterative learning algorithm is proposed for UDA using SLPP based subspace learning and the selective pseudo-labeling strategy.

– structured prediction is employed to explore the structural information within the target domain to promote the accuracy of pseudo-labeling and domain alignment.

– thorough comparative experiments and ablation studies are conducted to demonstrate the proposed approach can achieve new state-of-the-art performance on four benchmark datasets.

Related Work

Early works on UDA problems aim to align the marginal distributions of source and target domains (Gong et al. 2012; Ganin and Lempitsky 2015; Sun, Feng, and Saenko 2016). With aligned marginal distributions, it is not guaranteed to produce good classification results as the conditional distribution of the target domain can be misaligned with that of the source domain. This is mainly due to the lack of labeled target samples. To overcome this issue, many UDA approaches have employed pseudo-labeling strategies during learning (Long et al. 2013; Zhang, Li, and Ogunbona 2017; Wang et al. 2018; Pei et al. 2018; Zhang et al. 2018; Wang, Bu, and Breckon 2019; Chen et al. 2019b). Pseudo labeling the target samples allows to align the conditional distributions of source and target domains with traditional supervised learning algorithms. To give a sketch of how existing works handle the issue of lacking labeled samples in the target domain, in this section, we review related works on UDA by dividing them into three categories: *approaches without pseudo-labeling*, *pseudo-labeling without selection* and *pseudo-labeling with selection*.

Approaches without Pseudo-Labeling

Approaches to UDA aiming to align the marginal distributions of source and target domains can be realized via minimising the Maximum Mean Discrepancy (MMD) (Long et al. 2014; Sun, Feng, and Saenko 2016). The same idea has also been employed in deep learning based approaches to learning domain invariant features (Long et al. 2015; Sun and Saenko 2016; Long et al. 2016; Chen et al. 2019a). Alternatively, the same goal can be achieved by the gradient reversal layer (Ganin and Lempitsky 2015; Ganin et al. 2016) or generative adversarial loss (Tzeng et al. 2017). Although these models can learn domain invariant features which are also discriminative for the source domain, the separability of target samples is not guaranteed since the conditional distributions are not explicitly aligned. More recently, domain-symmetric networks were proposed to promote the alignment of joint distributions of feature and category across source and target domains (Zhang et al. 2019). In contrast to these approaches, pseudo-labeling target samples is another effective way to promote the alignment of conditional distributions.

Pseudo-Labeling without Selection

Pseudo-labeling without selection assigns pseudo-labels to all samples in the target domain. Two strategies, i.e. hard labeling (Long et al. 2013; Zhang, Li, and Ogunbona 2017; Wang et al. 2018) and soft labeling (Pei et al. 2018), have been employed in existing works. The strategy of hard labeling assigns a pseudo-label $\hat{y}$ to each unlabeled sample without considering the confidence. It can be achieved by a classifier trained on labeled source samples (Long et al. 2013; Zhang, Li, and Ogunbona 2017; Wang et al. 2018). The pseudo-labeled target samples together with labeled source samples are used to learn an improved classification model. By iterative learning, the pseudo-labeling is expected to be progressively more accurate until convergence. The problem of such hard pseudo-labeling is that mis-labeled samples by a weak classifier in the initial stage of the iterative learning can cause serious harm to the subsequent learning process. To address this issue, soft labeling was employed in (Pei et al. 2018). The strategy of soft labeling assigns the conditional probability of each class $p(c|x)$ given a target sample x, which results in a pseudo-labeling vector $\hat{y} = \{p(c_1|x), p(c_2|x), ..., p(c_{|\mathcal{C}|}|x)\} \in [0,1]^{|\mathcal{C}|}$, where $|\mathcal{C}|$ is the number of classes. The soft label $\hat{y}$ can be updated during iterative learning and the final classification results can be derived by selecting the class with the highest probability. Soft labeling is naturally suitable for neural network based approaches whose outputs are usually a vector of conditional probabilities. For instance, in the Multi-Adversarial Domain Adaptation (MADA) approach (Pei et al. 2018), the soft pseudo-label of a target sample is used to determine how much this sample should be attended to different class-specific domain discriminators.

Pseudo-Labeling with Selection

Selective pseudo-labeling is the other way to alleviate the mis-labeling issue (Zhang et al. 2018; Wang, Bu, and Breckon 2019; Chen et al. 2019b). Similar to the soft labeling strategy, selective pseudo-labeling also takes into consideration the confidence in target sample labeling but in a different manner. Specifically, a subset of target samples are selected to be assigned with pseudo labels and only these pseudo-labeled target samples are combined with source samples in the next iteration of learning. The idea is that at the beginning the classifier is weak so that only a small fraction of the target samples can be correctly classified. When the classifier gets stronger after each iteration of learning, more target samples can be correctly classified hence should be pseudo-labeled and participate in the learning process. One key factor in such algorithms is the criterion of sample selection for pseudo-labeling. An easy-to-hard strategy was employed in (Chen et al. 2019b). Target samples whose similarity scores are higher than a threshold are selected for pseudo-labeling and this threshold is updated after each iteration of learning so that more unlabeled target samples can be selected. One limitation of this sample selection strategy is the risk of biasing to "easy" classes and the selected samples in the first iterations can be dominated by these "easy" classes. As a result, the learned model will be seriously biased to the "easy" classes. To address this issue, a class-wise sample selection strategy was proposed in (Wang, Bu, and Breckon 2019). Samples are selected for each class independently so that pseudo-labeled target samples will contribute to the alignment of conditional distribution for each class during learning. In this paper, we propose a novel approach to selective pseudo-labeling by exploring the structural information within the unlabeled target samples.

Proposed Method

The proposed method aims to align the conditional distributions of source and target domains. We employ Supervised Locality Preserving Projection (SLPP) (He and Niyogi 2004) as an enabling technique to learn a projection matrix

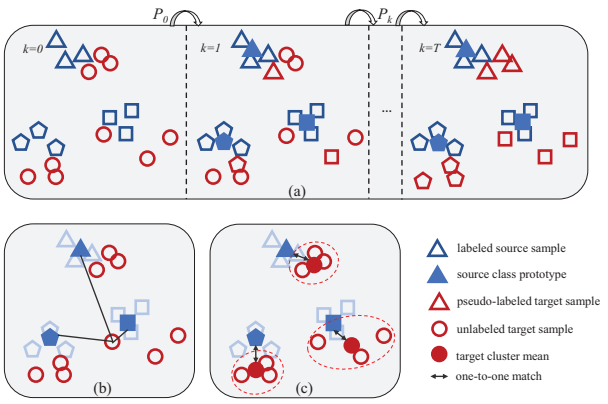

Figure 1: The framework of our proposed approach. (a) iterative learning process (b) pseudo-labeling via nearest class prototype (c) pseudo-labeling via structured prediction. The red and the blue colors are used for the target and source domains respectively.

$\mathbf{P}$ which maps samples from both domains into the same latent subspace. The learned subspace is expected to have favourable properties that projections of samples from the same class will be close to each other regardless of which domain they are from. In the subspace, a classifier such as nearest neighbour is used to classify unlabeled target samples. By combining both pseudo-labeled target samples and labeled source samples, the projection matrix $\mathbf{P}$ can be updated. As a result, an iterative learning process is employed to improve the projection learning and pseudo-labeling alternately as illustrated in Figure 1 (a).

Following (Wang, Bu, and Breckon 2019), we use the nearest class prototype (NCP) for pseudo-labeling. Source class prototypes are computed by averaging projections of source samples from the same class in the same space. Target samples can be pseudo-labeled by measuring the distances to these class prototypes (see Figure 1(b)). This method overlooks the intrinsic structural information in the target domain, resulting in sub-optimal pseudo-labeling results. Therefore, we explore the structural information underlying the target domain via clustering analysis (e.g., K-means). The clusters of target samples are matched with source classes via structured prediction (Zhang and Saligrama 2016; Wang and Chen 2017) so that target samples can be labeled collectively according to which cluster they belong to (see Figure 1 (c)). We calculate the distances of target samples to cluster centers as the criterion for selective pseudo-labeling. The samples close to the cluster center are more likely to be selected for pseudo-labeling and participate in the projection learning in the next iteration of learning. In contrast to the existing sample selection strategies (Chen et al. 2019b; Wang, Bu, and Breckon 2019), the structured prediction based method tends to select samples far away from the source samples which enables a faster domain alignment. Moreover, since these two methods are intrinsically different from each other, a combination of two is expected to further benefit the learning process.

In the following subsections, we give the formulation of the problem, describe each component of the proposed method in detail and analyse the complexity of the algorithm.

Problem Formulation

Given a labeled dataset $\mathcal{D}^s = \{(\boldsymbol{x}_i^s, y_i^s)\}, i = 1, 2, ..., n_s$ from the source domain $\mathcal{S}$, $\boldsymbol{x}_i^s \in \mathbb{R}^d$ represents the feature vector of i-th labeled sample in the source domain, d is the feature dimension and $y_i^s \in \mathcal{Y}^s$ denotes the corresponding label. UDA aims to classify an unlabeled data set $\mathcal{D}^t = \{\boldsymbol{x}_i^t\}, i = 1, 2, ..., n_t$ from the target domain $\mathcal{T}$, where $\boldsymbol{x}_i^t \in \mathbb{R}^d$ represents the feature vector in the target domain. The target label space $\mathcal{Y}^t$ is equal to the source label space $\mathcal{Y}^s$. It is assumed that both the labeled source domain data $\mathcal{D}^s$ and the unlabeled target domain data $\mathcal{D}^t$ are available for model learning.

Dimensionality Reduction

High dimensional features contain redundant information and thus result in unnecessary computation. We apply dimensionality reduction to the high dimensional deep features as the preprocessing. Principle Component Analysis (PCA) is selected in our work since it has been successfully used in other existing UDA approaches (Long et al. 2013; 2014). Feature vectors of samples from source and target domains are concatenated as a matrix $\mathbf{X} = [\boldsymbol{x}_1^s, ..., \boldsymbol{x}_{n_s}^s, \boldsymbol{x}_1^t, ..., \boldsymbol{x}_{n_t}^t] \in \mathbb{R}^{d \times n}$, where $n = n_s + n_t$. The centering matrix is denoted as $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}$, where $\mathbf{1}$ is an $n \times n$ matrix of ones. The objective of PCA is:

$$\max_{\mathbf{V}^T\mathbf{V}=\mathbf{I}} tr(\mathbf{V}^T\mathbf{X}\mathbf{H}\mathbf{X}^T\mathbf{V}) \tag{1}$$

where tr$(\cdot)$ denotes the trace of a matrix.

The problem defined in Eq.(1) is equivalent to the following eigenvalue problem:

$$\mathbf{X}\mathbf{H}\mathbf{X}^T\boldsymbol{v} = \phi\boldsymbol{v}. \tag{2}$$

By solving the eigenvalue problem, we can have the PCA projection matrix $\mathbf{V} = [\boldsymbol{v}_1, ..., \boldsymbol{v}_{d_1}] \in \mathbb{R}^{d \times d_1}$ and the lower-dimensional features:

$$\tilde{\mathbf{X}} = \mathbf{V}^T\mathbf{X} \tag{3}$$

where $\tilde{\mathbf{X}} \in \mathbb{R}^{d_1 \times n}$ and $d_1 \leq d$ is the dimensionality of the feature space after applying PCA.

Since PCA is a linear feature transformation, $L2$ normalization is applied to each feature vector in $\tilde{\mathbf{X}}$ as $\tilde{\boldsymbol{x}} \leftarrow \tilde{\boldsymbol{x}}/||\tilde{\boldsymbol{x}}||_2$. The use of $L2$ normalization forces samples of both source and target domains distributed on the surface of the same hyper-sphere which helps to align data from different domains (Wang and Chen 2017). Our experimental results in this study also provide empirical evidence that such sample normalization is beneficial to superior performance.

Domain Alignment

The lower-dimensional feature space $\tilde{\mathcal{X}}$ learned by PCA and sample normalisation exhibit favourable properties of domain alignment. However, it is learned in an unsupervised

manner and is thus not sufficiently discriminative. To promote the class-wise alignment of two domains, we use the supervised locality preserving projection (He and Niyogi 2004) as an enabling technique to learn a domain invariant yet discriminative subspace $\mathcal{Z}$ from $\tilde{\mathcal{X}}$.

The objective of SLPP is to learn a projection matrix $\mathbf{P}$ by minimizing the following cost function :

$$\min_{\mathbf{P}} \sum_{i,j} ||\mathbf{P}^T \tilde{\boldsymbol{x}}_{\boldsymbol{i}} - \mathbf{P}^T \tilde{\boldsymbol{x}}_{\boldsymbol{j}}||_2^2 \mathbf{M}_{ij}, \tag{4}$$

where $\mathbf{P} \in \mathbb{R}^{d_1 \times d_2}$ and $d_2 \leq d_1$ is the dimensionality of the learned space; $\tilde{\boldsymbol{x}}_i$ is the i-th column of the labeled data matrix $\tilde{\mathbf{X}}^l \in \mathbb{R}^{d_1 \times (n_s + n_t')}$ and $\tilde{\mathbf{X}}^l$ is a collection of n_s labeled source data and n_t' selected pseudo-labeled target data. The similarity matrix $\mathbf{M} \in \mathbb{R}^{(n_s+n_t') \times (n_s+n_t')}$ is defined as follows:

$$\mathbf{M}_{ij} = \begin{cases} 1, & y_i = y_j, \\ 0, & otherwise. \end{cases} \tag{5}$$

The idea is that samples from the same class should be projected close to each other in the subspace regardless of which domain they are originally from. Eq (5) is a simplified version of MMD matrices used in (Long et al. 2013; Zhang, Li, and Ogunbona 2017; Wang et al. 2018) where the domain differentiation is reserved while we try to promote the domain invariance. Our definition of similarity matrix in Eq (5) also differs from that in the original LPP formulation where local structure of the samples is considered by defining the similarity value $\mathbf{M}_{ij}$ based on the distance between $\tilde{\boldsymbol{x}}_{\boldsymbol{i}}$ and $\tilde{\boldsymbol{x}}_{\boldsymbol{j}}$.

Following the treatment in (He and Niyogi 2004; Wang and Chen 2017), the objective can be rewritten as:

$$\max_{\mathbf{P}} \frac{tr(\mathbf{P}^T \tilde{\mathbf{X}}^l \mathbf{D} \tilde{\mathbf{X}}^{lT} \mathbf{P})}{tr(\mathbf{P}^T (\tilde{\mathbf{X}}^l \mathbf{L} \tilde{\mathbf{X}}^{lT} + \mathbf{I}) \mathbf{P})} \tag{6}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{M}$ is the laplacian matrix, $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = \sum_j \mathbf{M}_{ij}$ and the regularization term $tr(\mathbf{P}^T \mathbf{P})$ is added for penalizing extreme values in the projection matrix $\mathbf{P}$.

The problem defined in Eq.(6) is equivalent to the following generalized eigenvalue problem:

$$\tilde{\mathbf{X}}^l \mathbf{D} \tilde{\mathbf{X}}^{lT} \boldsymbol{p} = \lambda (\tilde{\mathbf{X}}^l \mathbf{L} \tilde{\mathbf{X}}^{lT} + \mathbf{I}) \boldsymbol{p}, \tag{7}$$

solving the generalized eigenvalue problem gives the optimal solution $\mathbf{P} = [\boldsymbol{p}_1, ..., \boldsymbol{p}_{d_2}]$ where $\boldsymbol{p}_1, ..., \boldsymbol{p}_{d_2}$ are the eigenvectors corresponding to the largest d_2 eigenvalues.

Learning the projection matrix $\mathbf{P}$ for domain alignment requires labeled samples from both source and target domains. To get pseudo-labels of target samples for projection learning, we describe pseudo-labeling methods via nearest class prototype and structured prediction respectively in the following sub-sections.

Pseudo-Labeling via Nearest Class Prototype (NCP)

Unlabeled target samples can be labeled in the learned subspace $\mathcal{Z}$ where the projections of source and target samples are computed by:

$$\boldsymbol{z}^s = \mathbf{P}^T \tilde{\boldsymbol{x}}^s, \quad \boldsymbol{z}^t = \mathbf{P}^T \tilde{\boldsymbol{x}}^t. \tag{8}$$

We sequentially apply centralisation (i.e. mean subtraction, $\boldsymbol{z} \leftarrow \boldsymbol{z} - \bar{\boldsymbol{z}}$, where $\bar{\boldsymbol{z}}$ is the mean of all source and target sample projections) and $L2$ normalisation to $\boldsymbol{z}$ to promote the separability of different classes in the space $\mathcal{Z}$.

The class prototype for class $y \in \mathcal{Y}$ is defined as the mean vector of the projected source samples whose labels are y, which can be computed by:

$$\bar{\boldsymbol{z}}_y^s = \frac{\sum_{i=1}^{n_s} \boldsymbol{z}_i^s \delta(y, y_i^s)}{\sum_{i=1}^{n_s} \delta(y, y_i^s)}, \tag{9}$$

where $\delta(y, y_i) = 1$ if $y = y_i$ and 0 otherwise. After applying $L2$ normalization to the class prototypes $\bar{\boldsymbol{z}}_y^s, y = 1, ..., |\mathcal{Y}|$, where $|\mathcal{Y}|$ denotes the number of classes, we can derive the conditional probability of a given target sample $\boldsymbol{x}^t$ belonging to class y:

$$p_1(y|\boldsymbol{x}^t) = \frac{\exp\left(-||\boldsymbol{z}^t - \bar{\boldsymbol{z}}_y^s||\right)}{\sum_{y=1}^{|\mathcal{Y}|} \exp\left(-||\boldsymbol{z}^t - \bar{\boldsymbol{z}}_y^s||\right)}. \tag{10}$$

Pseudo-Labeling via Structured Prediction (SP)

Pseudo-labeling via nearest class prototype does not consider the intrinsic structure of the target samples which provides useful information for target samples classification. To explore such structure information, we employ structured prediction for pseudo-labeling. Specifically, we use K-means to generate $|\mathcal{Y}|$ clusters over projection vectors $\boldsymbol{z}^t$ of all target samples. The cluster centers are initialised with class prototypes calculated by Eq. (9). Subsequently, we establish a one-to-one match between a cluster from the target domain and a class from the source domain so that the sum of distances of all the matched pairs of the cluster center and the class prototype is minimised. Let $\mathbf{A} \in \{0, 1\}^{|\mathcal{Y}| \times |\mathcal{Y}|}$ denote the one-to-one matching matrix where $\mathbf{A}_{ij} = 1$ indicates that the i-th target cluster is matched with the j-th source class. The optimisation problem can be formulated as follows:

$$\begin{aligned} &\min_{\mathbf{A}} \sum_{i=1}^{|\mathcal{Y}|} \sum_{j=1}^{|\mathcal{Y}|} \mathbf{A}_{ij} d(\bar{\boldsymbol{z}}_i^t, \bar{\boldsymbol{z}}_j^s) \\ &s.t. \quad \forall i, \sum_j \mathbf{A}_{ij} = 1; \forall j, \sum_i \mathbf{A}_{ij} = 1, \end{aligned} \tag{11}$$

where $\bar{\boldsymbol{z}}_i^t$ denotes the i-th cluster center in the target domain. This problem can be efficiently solved by linear programming according to (Zhang and Saligrama 2016; Wang and Chen 2017).

Let $\bar{\boldsymbol{z}}_y^t$ denote the cluster center corresponding to the class y, similar to Eq. (10), we can calculate the conditional probability of a given target sample $\boldsymbol{x}^t$ belonging to class y:

$$p_2(y|\boldsymbol{x}^t) = \frac{\exp\left(-||\boldsymbol{z}^t - \bar{\boldsymbol{z}}_y^t||\right)}{\sum_{y=1}^{|\mathcal{Y}|} \exp\left(-||\boldsymbol{z}^t - \bar{\boldsymbol{z}}_y^t||\right)}. \tag{12}$$

Iterative Learning with Selective Pseudo-Labeling (SPL)

We use an iterative learning strategy to learn the projection matrix $\mathbf{P}$ for domain alignment and improved pseudo-labeling for target samples alternately. Although either of

Algorithm 1 Unsupervised Domain Adaptation Using Selective Pseudo-Labeling

Input: Labeled source data set $\mathcal{D}^s = \{(\boldsymbol{x}_i^s, y_i^s)\}, i = 1, 2, ..., n_s$ and unlabeled target data set $\mathcal{D}^t = \{\boldsymbol{x}_i^t\}, i = 1, 2, ..., n_t$, dimensionality of PCA and SLPP subspace d_1 and d_2, number of iteration T.

Output: The projection matrix $\mathbf{P}$ and predicted labels $\{\hat{y}^t\}$ for target samples.

```
Initialize k = 0;
Dimensionality reduction by Eq. (3);
Learn the projection P_0 using only source data D^s;
Assign pseudo labels for all target data using Eq. (14);
while k < T do
    k ← k + 1;
    Select a subset of pseudo-labeled target data S_k ∈ D̂^t;
    Learn P_k using D^s and S_k;
    Update pseudo labels for all target data using Eq.(14).
end while
```

the two pseudo-labeling methods described above is able to provide useful pseudo-labeled target samples for projection learning in the next iteration, they are intrinsically different. Pseudo-labeling via nearest class prototype tends to output high probability to the samples close to the source data, whilst structured prediction is confident in samples close to the cluster center in the target domain regardless how far they are from the source domain. We advocate to take advantage of the complementarity of these two methods via a simple combination of Eq.(10) and Eq.(12) as follows:

$$p(y|\boldsymbol{x}^t) = \max\{p_1(y|\boldsymbol{x}^t), p_2(y|\boldsymbol{x}^t)\}. \tag{13}$$

As a result, the pseudo-label of a given target sample $\boldsymbol{x}^t$ can be predicted by:

$$\hat{y}^t = \arg\max_{y \in \mathcal{Y}} p(y|\boldsymbol{x}^t). \tag{14}$$

Now we have pseudo labels for all the target samples as well as the probability of these pseudo labels, denoted as a set of triplets $\hat{\mathcal{D}}^t = \{(\boldsymbol{x}_i^t, \hat{y}_i^t, p(\hat{y}_i^t|\boldsymbol{x}_i^t))\}, i = 1, ..., n_t$.

Instead of using all the pseudo-labeled target samples for the projection learning, we progressively select a subset $\mathcal{S}_k \subseteq \hat{\mathcal{D}}^t$ containing kn_t/T target samples in the k-th iteration, where T is the number of iterations of the learning process. One straight forward strategy is to select top kn_t/T samples with highest probabilities from $\hat{\mathcal{D}}^t$. However, this strategy has a risk of only selecting samples from specific classes while overlooking the other classes. To avoid this, we do the class-wise selection so that target samples pseudo-labeled as each class have the same opportunity to be selected. Specifically, for each class $c \in \mathcal{Y}$, we first pick out n_t^c target samples pseudo-labeled as class c from which we select top kn_t^c/T high-probability samples to form $\mathcal{S}_k$. The overall algorithm is summarized in Algorithm 1.

Computational Complexity

We analyse the computation complexity of our learning algorithm. The complexity of PCA is $\mathcal{O}(dn^2 + d^3)$. The complexity of SLPP is $\mathcal{O}(2d_1n^2 + d_1^3)$ which is repeated for T times and leads to approximately $\mathcal{O}(T(2d_1n^2 + d_1^3))$. Since the iterative learning contributes the most to the computation cost, a small value of $d_1 < d$ can make the learning process more efficient when n is not too big.

Experiments and Results

In this section, we describe our experiments on four commonly used domain adaptation datasets (i.e. Office+Caltech (Gong et al. 2012), Office31 (Saenko et al. 2010), ImageCLEF-DA (Caputo et al. 2014) and Office-Home (Venkateswara et al. 2017)). Our approach is firstly compared with state-of-the-art UDA approaches to evaluate its effectiveness. An ablation study is conducted to demonstrate the effects of different components and hyper-parameters in our approach. Finally, we investigate how different hyper-parameters affect the performance.

Datasets

Office+Caltech (Gong et al. 2012) consists of four domains: Amazon (A, images downloaded from online merchants), Webcam (W, low-resolution images by a web camera), DSLR (D, high-resolution images by a digital SLR camera) and Caltech-256 (C). Ten common classes from all four domains are used: backpack, bike, calculator, headphone, computer-keyboard, laptop-101, computer-monitor, computer-mouse, coffee-mug, and video-projector. There are 2533 images in total with 8 to 151 images per category per domain. **Office31** (Saenko et al. 2010) consists of three domains: Amazon (A), Webcam (W) and DSLR (D). There are 31 common classes for all three domains containing 4,110 images in total. **ImageCLEF-DA** (Caputo et al. 2014) consists of four domains. We follow the existing works (Zhang et al. 2019) using three of them in our experiments: Caltech-256 (C), ImageNet ILSVRC 2012 (I), and Pascal VOC 2012 (P). There are 12 classes and 50 images for each class in each domain. **Office-Home** (Venkateswara et al. 2017) is another dataset recently released for evaluation of domain adaptation algorithms. It consists of four different domains: Artistic images (A), Clipart (C), Product images (P) and Real-World images (R). There are 65 object classes in each domain with a total number of 15,588 images.

Experimental Setting

The algorithm is implemented in Matlab[1]. We use the deep features commonly used in existing works for a fair comparison with the state of the arts. As a result, the Decaf6 (Donahue et al. 2014) features (activations of the 6*th* fully connected layer of a convolutional neural network trained on ImageNet, $d = 4096$) are used for the Office-Caltech dataset. For the other three datasets, ResNet50 (He et al. 2016) has been commonly used to extract features or as the backbone of deep models in the literature, hence we

[1]Code is available: https://github.com/hellowangqian/domain-adaptation-capls

Table 1: Classification Accuracy (%) on Office-Caltech dataset using Decaf6 features. Each column displays the results of a pair of source → target setting.

Method	C→A	C→W	C→D	A→C	A→W	A→D	W→C	W→A	W→D	D→C	D→A	D→W	Average
DDC(Tzeng et al. 2014)	91.9	85.4	88.8	85.0	86.1	89.0	78.0	84.9	**100.0**	81.1	89.5	98.2	88.2
DAN(Long et al. 2015)	92.0	90.6	89.3	84.1	91.8	91.7	81.2	92.1	**100.0**	80.3	90.0	98.5	90.1
DCORAL(Sun and Saenko 2016)	92.4	91.1	91.4	84.7	-	-	79.3	-	-	82.8	-	-	-
CORAL(Sun, Feng, and Saenko 2017)	92.0	80.0	84.7	83.2	74.6	84.1	75.5	81.2	**100.0**	76.8	85.5	99.3	84.7
SCA(Ghifary et al. 2016)	89.5	85.4	87.9	78.8	75.9	85.4	74.8	86.1	**100.0**	78.1	90.0	98.6	85.9
JGSA(Zhang, Li, and Ogunbona 2017)	91.4	86.8	93.6	84.9	81.0	88.5	85.0	90.7	**100.0**	86.2	92.0	99.7	90.0
MEDA(Wang et al. 2018)	**93.4**	**95.6**	91.1	**87.4**	88.1	88.1	**93.2**	**99.4**	99.4	87.5	**93.2**	97.6	92.8
CAPLS (Wang, Bu, and Breckon 2019)	90.8	85.4	95.5	86.1	87.1	**94.9**	88.2	92.3	**100.0**	**88.8**	93.0	**100.0**	91.8
SPL (Ours)	92.7	93.2	**98.7**	**87.4**	**95.3**	89.2	87.0	92.0	**100.0**	88.6	92.9	98.6	**93.0**

Table 2: Classification Accuracy (%) on Office31 dataset using either ResNet50 features or ResNet50 based deep models.

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
RTN(Long et al. 2016)	84.5	96.8	99.4	77.5	66.2	64.8	81.6
MADA(Pei et al. 2018)	90.0	97.4	99.6	87.8	70.3	66.4	85.2
GTA (Sankaranarayanan et al. 2018)	89.5	97.9	99.8	87.7	72.8	71.4	86.5
iCAN(Zhang et al. 2018)	92.5	**98.8**	**100.0**	90.1	72.1	69.9	87.2
CDAN-E(Long et al. 2018)	94.1	98.6	**100.0**	92.9	71.0	69.3	87.7
JDDA(Chen et al. 2019a)	82.6	95.2	99.7	79.8	57.4	66.7	80.2
SymNets(Zhang et al. 2019)	90.8	**98.8**	**100.0**	**93.9**	74.6	72.5	88.4
TADA (Wang et al. 2019)	**94.3**	98.7	99.8	91.6	72.9	73.0	88.4
MEDA(Wang et al. 2018)	86.2	97.2	99.4	85.3	72.4	74.0	85.7
CAPLS (Wang, Bu, and Breckon 2019)	90.6	98.6	99.6	88.6	75.4	76.3	88.2
SPL (Ours)	92.7	98.7	99.8	93.0	**76.4**	**76.8**	**89.6**

Table 3: Classification Accuracy (%) on ImageCLEF-DA dataset using either ResNet50 features or ResNet50 based deep models.

Method	I→P	P→I	I→C	C→I	C→P	P→C	Avg
RTN(Long et al. 2016)	75.6	86.8	95.3	86.9	72.7	92.2	84.9
MADA(Pei et al. 2018)	75.0	87.9	96.0	88.8	75.2	92.2	85.8
iCAN(Zhang et al. 2018)	79.5	89.7	94.7	89.9	78.5	92.0	87.4
CDAN-E(Long et al. 2018)	77.7	90.7	**97.7**	91.3	74.2	94.3	87.7
SymNets(Zhang et al. 2019)	**80.2**	93.6	97.0	93.4	78.7	**96.4**	89.9
MEDA(Wang et al. 2018)	79.7	92.5	95.7	92.2	78.5	95.5	89.0
SPL (Ours)	78.3	**94.5**	96.7	**95.7**	**80.5**	96.3	**90.3**

use ResNet50 features ($d = 2048$) in our experiments. Although a small dimensionality of the PCA space d_1 is preferred for less computation, it can cause information loss for a dataset with a large number of classes (e.g., Office-Home). To trade off, we set the values of d_1 based on the number of classes in the dataset which results in $d_1 = 128, 512, 128$ and 1024 for Office-Caltech, Office-31, ImageCLEF-DA and Office-Home respectively. For the dimensionality of the space learned by SLPP, we set $d_2 = 128$ uniformly for all datasets. The number of iterations T is set to 10 in all experiments unless otherwise specified.

Comparison with State-of-the-Art Approaches

We compare our approach with the state of the arts including those based on deep features (extracted using deep models such as ResNet50 pre-trained on ImageNet) and deep learning models. The classification accuracy of our approaches and the comparative ones are shown in Tables 1-4 in terms of each combination of "source" → "target" domains and the average accuracy over all different combinations. In each table, the results of deep learning based models are listed on the top followed by deep feature based methods including ours. Our approach is denoted as SPL (Selective Pseudo-Labeling). We use bold and underlined fonts to indicate the best and the second best results respectively in each setting.

From Tables 1-4 we can see that our proposed approach with the combination of two different pseudo-labeling methods achieves the highest average accuracy (see the last column of each table) consistently on four datasets. Specifically, our proposed SPL achieves an average accuracy of 93.0% on the Office-Caltech dataset (Table 1), slightly better than MEDA (Wang et al. 2018) which has an average accuracy of 92.8%. On the Office31 dataset (Table 2), SPL achieves the best or the second-best performance in five out of six tasks and the best average performance of 89.6% while the second-highest average accuracy (88.4%) was achieved by the deep learning based approaches SymNets (Zhang et al. 2019) and (Wang et al. 2019). On the ImageCLEF-DA dataset (Table 3), the proposed SPL approach performs the best or the second-best in four out of six tasks and ranks the first with the average accuracy of 90.5% followed by the deep learning model SymNets (Zhang et al. 2019, 89.9%) and deep feature based model MEDA (Wang et al. 2018, 89.0%). On the Office-Home dataset (Table 4, again, our approach SPL outperforms all state-of-the-art models with an average accuracy of 71.0% against 70.6% by CAPLS (Wang, Bu, and Breckon 2019) and 67.6% by SymNets (Zhang et al. 2019) and TADA (Wang et al. 2019).

In summary, our selective pseudo-labeling approach can outperform both deep learning models and traditional feature transformation approaches on four commonly used datasets for UDA.

Ablation Study

We conduct an ablation study to analyse how different components of our work contribute to the final performance. To this end, we investigate different combinations of four components: pseudo-labeling (PL), sample selection (S) for pseudo-labeling, nearest class prototype (NCP) and structured prediction (SP). We report the average classification accuracy on four datasets in Table 5. It can be observed that methods with pseudo-labeling outperform those without pseudo-labeling and the use of selective pseudo-labeling further improves the performance significantly on all datasets. In terms of the pseudo-labeling strategy, structured prediction (SP) outperforms nearest class prototype (NCP) consistently while the combination of these two can further improve the accuracy marginally.

Table 4: Classification Accuracy (%) on Office-Home dataset using either ResNet50 features or ResNet50 based deep models.

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Average
JAN(Long et al. 2017)	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN-E (Long et al. 2018)	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
SymNets (Zhang et al. 2019)	47.7	72.9	78.5	64.2	71.3	74.2	64.2	48.8	79.5	**74.5**	52.6	82.7	67.6
TADA (Wang et al. 2019)	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	**60.0**	82.9	67.6
MEDA(Wang et al. 2018)	54.6	75.2	77.0	56.5	72.8	72.3	59.0	51.9	78.2	67.7	57.2	81.8	67.0
CAPLS (Wang, Bu, and Breckon 2019)	**56.2**	**78.3**	80.2	**66.0**	75.4	78.4	**66.4**	**53.2**	81.1	71.6	56.1	84.3	70.6
SPL (Ours)	54.5	77.8	**81.9**	65.1	**78.0**	**81.1**	66.0	53.1	**82.8**	69.9	55.3	**86.0**	**71.0**

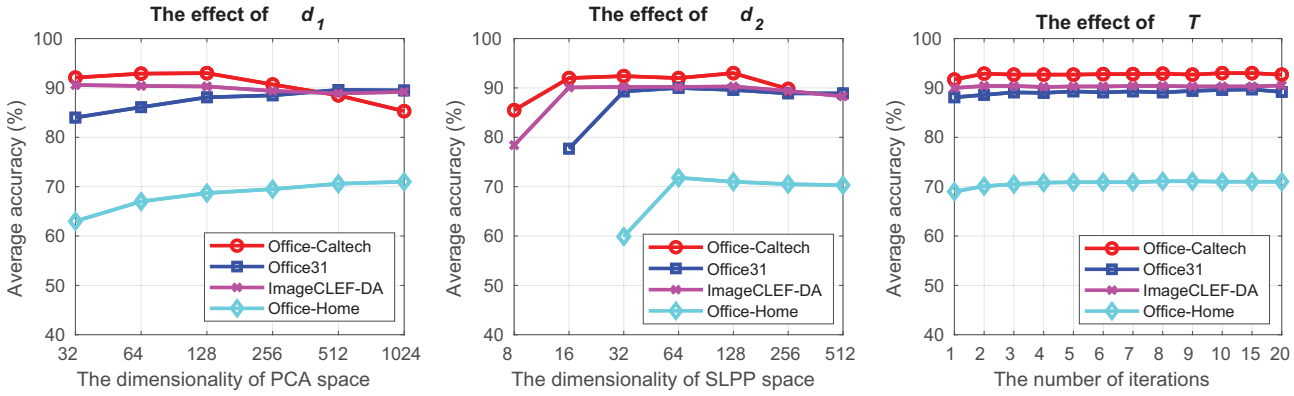

Figure 2: The effect of hyper-parameters (i.e. the dimensionality of PCA space d_1, the dimensionality of SLPP sapce d_2 and the number of iterations T).

Table 5: Results of ablation study.

Method				Office-Caltech	Office31	ImageCLEF-DA	Office-Home
PL	S	NCP	SP				
✗	✗	✓	✗	81.8	82.0	86.2	63.9
✗	✗	✗	✓	90.3	87.5	89.5	68.0
✗	✗	✓	✓	90.7	87.6	89.4	68.1
✓	✗	✓	✗	85.5	83.7	86.9	66.2
✓	✗	✗	✓	91.9	88.0	90.0	68.9
✓	✗	✓	✓	92.0	88.0	90.0	69.0
✓	✓	✓	✗	90.8	87.8	89.0	70.8
✓	✓	✗	✓	93.0	89.5	90.2	71.0
✓	✓	✓	✓	93.0	89.6	90.3	71.0

Effects of Hyper-parameters

Our approach has three hyper-parameters: the dimensionality of PCA space d_1, the dimensionality of SLPP space d_2 and the number of iterations T. We investigate how each hyper-parameter affects the performance by setting it to a series of different values while fixing the other two. The results are shown in Figure 2 in which the average accuracy over all possible source-target pairs are reported for four datasets. As we can see, the datasets with more classes (e.g., Office31 and Office-Home) tend to large values of d_1 while small values of d_1 are beneficial to the datasets with fewer classes (e.g., Office-Caltech and ImageCLEF-DA). In terms of the dimensionality of SLPP space d_2, the performance does not change too much unless its value is less than the number of classes. The number of iterations T has nearly zero effect on the performance when it is greater than 2. To summarize, our approach is not sensitive to the hyper-parameters and performs comparably well if only d_2 is set greater than the number of classes.

Conclusion

We propose a novel selective pseudo-labeling approach to UDA by incorporating supervised subspace learning and structured prediction based pseudo-labeling into an iterative learning framework. The proposed approach outperforms other state-of-the-art methods on four benchmark datasets. The ablation study demonstrates the effectiveness of selective pseudo-labeling and structured prediction which can also be employed to train the deep learning models for UDA in the future work.

References

Caputo, B.; Müller, H.; Martinez-Gomez, J.; Villegas, M.; Acar, B.; Patricia, N.; Marvasti, N.; Üsküdarlı, S.; Paredes, R.; Cazorla, M.; et al. 2014. Imageclef 2014: Overview and analysis of the results. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, 192–211. Springer.

Chen, C.; Chen, Z.; Jiang, B.; and Jin, X. 2019a. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *AAAI Conference on Artificial Intelligence*.

Chen, C.; Xie, W.; Huang, W.; Rong, Y.; Ding, X.; Huang, Y.; Xu, T.; and Huang, J. 2019b. Progressive feature alignment for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 627–636.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image

database. In *IEEE conference on computer vision and pattern recognition*, 248–255. IEEE.

Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; and Darrell, T. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning*, 647–655.

Ganin, Y., and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, 1180–1189.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* 17(1):2096–2030.

Ghifary, M.; Balduzzi, D.; Kleijn, W. B.; and Zhang, M. 2016. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(7):1414–1430.

Gong, B.; Shi, Y.; Sha, F.; and Grauman, K. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2066–2073. IEEE.

He, X., and Niyogi, P. 2004. Locality preserving projections. In *Advances in neural information processing systems*, 153–160.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition*, 770–778.

Long, M.; Wang, J.; Ding, G.; Sun, J.; and Yu, P. S. 2013. Transfer feature learning with joint distribution adaptation. In *International Conference on Computer Vision*, 2200–2207.

Long, M.; Wang, J.; Ding, G.; Sun, j.; and Yu, P. S. 2014. Transfer joint matching for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition*, 1410–1417.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. I. 2015. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, 97–105. JMLR. org.

Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2016. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, 136–144.

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

Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2018. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, 1647–1657.

Pei, Z.; Cao, Z.; Long, M.; and Wang, J. 2018. Multi-adversarial domain adaptation. In *AAAI Conference on Artificial Intelligence*.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. In *European Conference on Computer Vision*, 213–226. Springer.

Sankaranarayanan, S.; Balaji, Y.; Castillo, C. D.; and Chellappa, R. 2018. Generate to adapt: Aligning domains using generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Sun, B., and Saenko, K. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, 443–450. Springer.

Sun, B.; Feng, J.; and Saenko, K. 2016. Return of frustratingly easy domain adaptation. In *AAAI 2016*, volume 6, 8.

Sun, B.; Feng, J.; and Saenko, K. 2017. Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*. 153–171.

Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; and Darrell, T. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*.

Tzeng, E.; Hoffman, J.; Saenko, K.; and Darrell, T. 2017. Adversarial discriminative domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, 4.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 5018–5027.

Wang, Q., and Chen, K. 2017. Zero-shot visual recognition via bidirectional latent embedding. *International Journal of Computer Vision* 124(3):356–383.

Wang, J.; Feng, W.; Chen, Y.; Yu, H.; Huang, M.; and Yu, P. S. 2018. Visual domain adaptation with manifold embedded distribution alignment. In *2018 ACM Multimedia Conference on Multimedia Conference*, 402–410. ACM.

Wang, X.; Li, L.; Ye, W.; Long, M.; and Wang, J. 2019. Transferable attention for domain adaptation. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Wang, Q.; Bu, P.; and Breckon, T. P. 2019. Unifying unsupervised domain adaptation and zero-shot visual recognition. In *International Joint Conference on Neural Networks*.

Zhang, Z., and Saligrama, V. 2016. Zero-shot recognition via structured prediction. In *European conference on computer vision*, 533–548. Springer.

Zhang, W.; Ouyang, W.; Li, W.; and Xu, D. 2018. Collaborative and adversarial network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3801–3809.

Zhang, Y.; Tang, H.; Jia, K.; and Tan, M. 2019. Domain-symmetric networks for adversarial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5031–5040.

Zhang, J.; Li, W.; and Ogunbona, P. 2017. Joint geometrical and statistical alignment for visual domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 5150–5158. IEEE.