Unsupervised Domain Adaptation with Imbalanced Cross-Domain Data

Tzu-Ming Harry Hsu[1], Wei-Yu Chen[1], Cheng-An Hou[2], Yao-Hung Hubert Tsai[3],
Yi-Ren Yeh[4], and Yu-Chiang Frank Wang[3]
[1]Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
[2]The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[3]Research Center for IT Innovation, Academia Sinica, Taipei, Taiwan
[4]Department of Mathematics, National Kaohsiung Normal University, Kaohsiung, Taiwan
{harry19930924, wyharveychen, c.a.andyhou, y.h.huberttsai}@gmail.com, yryeh@nknu.edu.tw, ycwang@citi.sinica.edu.tw

Abstract

We address a challenging unsupervised domain adaptation problem with imbalanced cross-domain data. For standard unsupervised domain adaptation, one typically obtains labeled data in the source domain and only observes unlabeled data in the target domain. However, most existing works do not consider the scenarios in which either the label numbers across domains are different, or the data in the source and/or target domains might be collected from multiple datasets. To address the aforementioned settings of imbalanced cross-domain data, we propose Closest Common Space Learning (CCSL) for associating such data with the capability of preserving label and structural information within and across domains. Experiments on multiple cross-domain visual classification tasks confirm that our method performs favorably against state-of-the-art approaches, especially when imbalanced cross-domain data are presented.

1. Introduction

For pattern recognition problems, one typically trains the classifiers using pre-collected training data, aiming at recognizing test instances which are not seen during training. This implies that training and test data exhibit similar data or feature distributions. However, in real-world applications, training and test data might be collected by different users, using distinct sensors, at dissimilar scenarios, or during separate time periods. Such data are considered to be present in different *domains*, and the difference between them (or the mismatch between their data distributions) is thus not negligible.

Domain adaptation addresses the tasks in which training and test data are collected from *source* and *target* domains, respectively. Its goal is to eliminate domain differences for relating cross-domain data. Depending on the availability of labeled data in the target domain during training, one can generally divide existing techniques into two categories: *semi-supervised* and *unsupervised* domain adaptation.

For semi-supervised domain adaptation, either a small number of target-domain labeled data or cross-domain data pairs can be observed during training [21]. For example, Jiang and Zhai [15] apply instance reweighting techniques for adapting classifiers learned from source to target domains. By utilizing the correspondence information across source and target domains, Huang and Wang [12] advance dictionary learning to derive a common feature space, which can be applied for the tasks of cross-domain classification and synthesis. As noted in [23], adaptation problems with significant domain differences or distribution mismatches (e.g., pose-invariant face recognition [23] or image-to-text classification [28]) generally require semi-supervised settings for achieving satisfactory performance.

For unsupervised domain adaptation, one can collect labeled data in the source domain, while only unlabeled data to be recognized can be observed in the target domain during training (no cross-domain instance pair is available either). Since there is no label information available in the target domain, how to transfer such information from the source domain becomes a challenging task. Based on *Maximum Mean Discrepancy* (MMD) [9], recent approaches choose to eliminate the domain difference by matching cross-domain data distributions [30, 24, 20, 17]. As discussed in Section 2, the basic idea of such methods is to derive a common feature space, in which the marginal and/or conditional distributions of cross-domain can be matched.

However, existing approaches for unsupervised domain adaptation typically assume that the label numbers of the source and target domains are the same (e.g., [20, 18, 17]). They also expect that the data of each class presented in the source or target domains exhibit similar data distributions. In practice, the number of categories in the source domain is

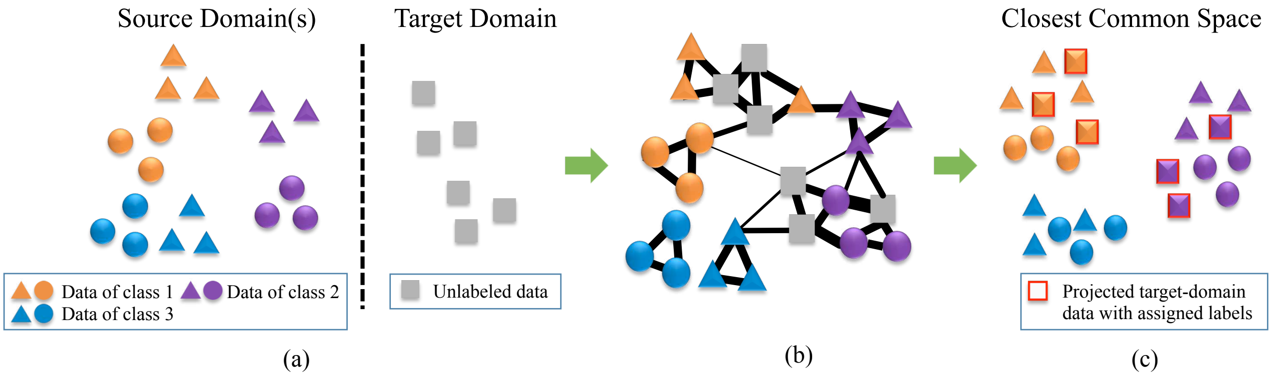


Figure 1: Overview of our approach: (a) Source and target-domain data, (b) exploiting label and latent-domain information within and across domains, and (c) the resulting space for adaptation and classification. Note that different shapes correspond to different domains/datasets, while the colors denote the object categories.

often larger than that in the target domain. Moreover, both source and target-domain data might be collected from multiple datasets. In this paper, we refer to the aforementioned scenarios as the presence of *imbalanced cross-domain data*.

While some researchers advocate instance selection or latent domain discovery [6, 7, 27] to handle problems with problems with mixed source-domain data, they cannot be easily applied for solving domain adaptation tasks in which the label numbers do not match across domains. In our work, we also propose an MMD-based algorithm of *Closest Common Space Learning* (CCSL). The major advantage of our CCSL is its ability in dealing with imbalanced cross-domain data for unsupervised domain adaptation. We will show that, by exploiting label and structural information within and across domains, latent source domains can be identified for adaptation and recognition purposes.

The contributions of this paper are summarized below:

- We propose a novel unsupervised domain adaptation algorithm of *Closest Common Space Learning* (CCSL), which jointly solves instance reweighting and subspace learning to learn the latent sub-domains for adaptation. (Section 3)
- Our CCSL exploits both label and structural information for data within and across domains. This is achieved by relating latent source-target domain pairs, with the ability to disregard irrelevant source domain instances during adaptation. (Section 3)
- In addition to achieving satisfactory performance on benchmark cross-domain classification datasets, our method is able to perform favorably against recent unsupervised domain adaptation approaches on problems with imbalanced cross-domain data. (Section 4)

2. Related Works

In this section, we briefly review recent works on unsupervised domain adaptation. Generally, one can divide existing approaches into three categories: *instance reweighting* [13, 25], *feature space matching* [20, 8, 5, 17, 6], and *latent domain discovery* [11, 7]. Viewing the importance or contribution of each source-domain instance different during adaptation, instance reweighting suppresses the difference between source and target domain data by minimizing the MMD [9] or the *Kullback-Leibler distances* [25]. Classification-based methods like [3] apply selected source-domain classifiers to recognize the matched target-domain instances. Nevertheless, reweighting the source-domain data might be not sufficient for adapting cross-domain data, if the domain difference is not simply a domain shift.

Feature space matching is among the popular techniques for unsupervised domain adaptation. Such strategies aim at discovering a common feature space which allows matching of data distributions across domains. For example, Pan *et al*. [20] proposed *Transfer Component Analysis* (TCA) to project cross-domain data into low dimensional embeddings for matching their marginal distributions. Long *et al*. developed [17] *Joint Distribution Adaptation* (JDA), which adapts both marginal and conditional data distributions when deriving the common feature space. Different from MMD-based approaches, Gong *et al*. [8, 6] constructed a Riemannian manifold and defined *Geodesic Flow Kernel* (GFK) for matching cross-domain data. Similarly, Baktashmotlagh *et al*. [1, 2] applied manifold learning to achieve the above goal by minimizing the *Hellinger distance* between cross-domain data distributions. Dictionary-learning based approaches methods like [19, 29] can also be considered in this category. With the same goal of associating cross-domain data, they adapt the source-domain dictionary to the target domain by observing the data in that domain accordingly.

To deal with data collected from more than one domain, latent domain discovery decomposes the observed source or target-domain data into multiple sub-domains for improved adaptation. For example, Hoffman *et al*. [11] chose to cluster the source-domain data with constraints on their label information. To minimize the MMD between the source-domain data in different sub-domains, Gong *et al*. [7] pursued their maximally distinctive distributions. Recently, Xu *et al*. [27] utilized exemplar SVMs to identify multiple sub-domains for source-domain data via low-rank approximation. However, once the sub-domains are determined, the aforementioned works simply select the one closest to the target domain for adaptation. Moreover, existing latent domain discovery approaches typically assume that the label numbers are the same across domains, which would also limit their practical uses.

3. Our Proposed Method

3.1. Problem Settings

We first define the problem formulation and introduce the notations which will be used in this paper. Let the training data in the source domain as $\mathcal{D}_S = \left\{\left(\mathbf{x}_i^S, y_i^S\right)\right\}_{i=1}^{N_S} = \{\mathbf{X}_S, \mathbf{y}_S\}$, where $\mathbf{X}_S \in \mathbb{R}^{l\times N_S}$ denotes N_S l-dimensional source-domain data, and each entry y_i^S in $\mathbf{y}_S \in \mathbb{R}^{N_S}$ indicates the corresponding label of C categories. As for the target domain, the unlabeled data are represented using the same type of features. Thus, we have $\mathcal{D}_T = \left\{\left(\mathbf{x}_j^T, y_j^T\right)\right\}_{j=1}^{N_T} = \{\mathbf{X}_T, \mathbf{y}_T\}$, where $\mathbf{X}_T \in \mathbb{R}^{l\times N_T}$ is the observed target-domain data, and $\mathbf{y}_T \in \mathbb{R}^{N_T}$ is the label vector to be determined.

It is worth repeating that, for unsupervised domain adaptation with imbalanced cross-domain data, we not only deal with possible mixed source or target domain data (i.e., instances in $\mathbf{X}_S$ or $\mathbf{X}_T$ of the same class but collected from different datasets). We also consider that the label number C of the source domain might be larger than or equal to that in the target domain.

3.2. Beyond Matching Cross-Domain Marginal and Conditional Distributions

Recall that, to eliminate the domain differences, JDA determines a feature transformation $\Phi(\cdot)$, which projects source and target domain data to a common subspace for matching cross-domain marginal and conditional data distributions. In other words, the goal of JDA is to satisfy $\mathcal{P}_S(\phi(\mathbf{X}_S)) \approx \mathcal{P}_T(\phi(\mathbf{X}_T))$ and $\mathcal{P}_S(\phi(\mathbf{X}_S)|\mathbf{y}_S) \approx \mathcal{P}_T(\phi(\mathbf{X}_T)|\mathbf{y}_T)$ by minimizing the following MMD distance $\mathcal{M}_\phi$:

$$
\begin{aligned}
&\mathcal{M}_\phi\left(\mathcal{P}_S(\mathbf{X}_S, \mathbf{y}_S), \mathcal{P}_T(\mathbf{X}_T, \mathbf{y}_T)\right)\\
&\approx \mathcal{M}_\phi\left(\mathcal{P}_S(\mathbf{X}_S), \mathcal{P}_T(\mathbf{X}_T)\right) + \mathcal{M}_\phi\left(\mathcal{P}_S(\mathbf{X}_S|\mathbf{y}_S), \mathcal{P}_T(\mathbf{X}_T|\mathbf{y}_T)\right).
\end{aligned}
$$

Since only unlabeled data can be observed in the target domain, JDA applies source-domain classifiers to predict the *pseudo labels* of the target-domain data, which allows the matching of cross-domain conditional data distributions for adaptation purposes.

Despite promising performance, JDA and most MMD-based approaches regard each data domain as an atomic distribution. In practice, source or target-domain data can be collected by different users using distinct sensors, and thus there would exist latent sub-domains for the collected data. Moreover, the number of categories in the source domain might be larger than that in the target domain.

To address unsupervised domain adaptation with imbalanced cross-domain data, we propose a novel algorithm of *Closest Common Space Learning* (CCSL). Instead of assuming that the data in each domain exhibit atomic distributions, our CCSL considers a latent domain variable d for exploiting both label and structural information within and across domains during adaptation. Thus, our CCSL aims at minimizing the following MMD distance $\mathcal{M}_{\phi,d}$ (d denotes domain-dependent MMD) :

$$
\begin{aligned}
&\mathcal{M}_{\phi,d}\left(\mathcal{P}_S(\mathbf{X}_S, \mathbf{y}_S), \mathcal{P}_T(X_T, \mathbf{y}_T)\right)\\
\approx &\mathcal{M}_\phi\left(\mathcal{P}_S(\mathbf{X}_S), \mathcal{P}_T(\mathbf{X}_T)\right)\\
&+ \mathcal{M}_{\phi,d}\left(\mathcal{P}_S(\mathbf{X}_S|\mathbf{y}_S), \mathcal{P}_T(\mathbf{X}_T|\mathbf{y}_T)\right).
\end{aligned} \tag{1}
$$

The first term in (1) denotes the matching of cross-domain marginal distributions, and the second term takes both label and latent-domain information for matching cross-domain conditional distributions. That is, different from matching cross-domain conditional distributions using class means with pseudo labels, we propose to exploit both label and latent structure similarities within and across domains for adaptation. This is achieved by jointly solving the tasks of instance reweighting and subspace learning in a unified framework, as detailed in the following subsection.

3.3. Closest Common Space Learning

As noted in Section 3.2, we propose to exploit both label and latent-domain information within and across domains for matching cross-domain conditional distribution. More specifically, we define the second term in (1) as:

$$
\begin{aligned}
&\mathcal{M}_{\phi,d}\left(\mathcal{P}_S(\mathbf{X}_S|\mathbf{y}_S), \mathcal{P}_T(\mathbf{X}_T|\mathbf{y}_T)\right)\\
&= \sum_{i,j} \frac{m_{ij}^{ST}}{\sum_k m_{ki}^{SS} \sum_l m_{lj}^{TT}} \left\|\hat{\phi}(\mathbf{x}_i^S) - \hat{\phi}(\mathbf{x}_j^T)\right\|^2,
\end{aligned} \tag{2}
$$

where

$$
\mathbf{M} = \begin{bmatrix} \mathbf{M}^{SS} & \mathbf{M}^{ST} \\ \mathbf{M}^{TS} & \mathbf{M}^{TT} \end{bmatrix} \in \mathbb{R}^{(N_S+N_T)\times(N_S+N_T)}
$$

and

$$\hat{\phi}(\mathbf{x}_i^S) = \frac{\sum_l m_{li}^{SS} \phi(\mathbf{x}_l^S)}{\sum_l m_{li}^{SS}}, \hat{\phi}(\mathbf{x}_j^T) = \frac{\sum_l m_{lj}^{TT} \phi(\mathbf{x}_l^T)}{\sum_l m_{lj}^{TT}}.$$

In (2), the similarity matrix $\mathbf{M} \in \mathbb{R}^{(N_S+N_T)\times(N_S+N_T)}$ associates each within and cross-domain data pair. Each entry m_{ij}^{ST} in the cross-domain similarity matrix $\mathbf{M}^{ST}$ measures the label and latent-domain similarities for each cross-domain data pair, while m_{li}^{SS} and m_{lj}^{TT} exploit the latent structures for the associated data within source and target domains, respectively (see Section 3.3.1 for the derivation of $\mathbf{M}$). As a result, minimizing (2) is equivalent to the matching of cross-domain data distributions based on the conditions of the observed labels and latent domains.

3.3.1 Observing label and latent-domain similarities

We now explain how we determine $\mathbf{M}$ in (2). Given labeled source-domain and unlabeled target-domain data, we apply a set of *linear discriminators* $\mathbf{w}_i$, each is trained by a source or target-domain instance of interest in the resulting feature space (via ϕ). Thus, we have $\mathbf{W} = [\mathbf{w_1}, ..., \mathbf{w}_{N_S}, ...\mathbf{w}_{N_S+N_T}] \in \mathbb{R}^{k\times(N_S+N_T)}$, where k indicates the dimensions of our closest common space. To learn each $\mathbf{w}_i$, we follow the strategies below:

- If $\mathbf{w}_i$ is trained by a projected source-domain instance $\phi(\mathbf{x}_i)$, we take a portion p $(0 \leqslant p \leqslant 1)$ of the projected data with the same label as $\mathbf{x}_j$ as positive instances (selected by nearest neighbors), while the remaining ones with distinct labels will viewed as negative samples.

- If $\mathbf{w}_i$ is trained by a projected target-domain instance $\phi(\mathbf{x}_i)$, we follow JDA and apply source-domain SVMs to predict its pseudo label $\hat{y}_i^T$. The procedure of selecting positive and negative samples to train $\mathbf{w}_i$ for $\mathbf{x}_i$ is the same as the case above.

Once $\mathbf{w}_i$ for each instance is derived, we apply them to predict the output scores p for each instance $\mathbf{x}_j$, which is in the same or different domain as $\mathbf{x}_i$ is. Finally, this score will be normalized to $[0, 1]$ as the corresponding entry in $\mathbf{M}$ using a sigmoid function $\sigma(g) = 1/(1 + e^{-g})$, where $g = \mathbf{w}_i^T \phi(\mathbf{x}_j)$. Once the similarity matrices of $\mathbf{M}^{SS}$, $\mathbf{M}^{TT}$, and $\mathbf{M}^{ST}$ are determined, $\hat{\phi}(\mathbf{x}_i^S)$ and $\hat{\phi}(\mathbf{x}_j^T)$ can be derived based on their definitions in (2).

It can be seen that, instead of measuring the difference between cross-domain instance pairs, the use of $\hat{\phi}(\mathbf{x}_i^S)$ and $\hat{\phi}(\mathbf{x}_j^T)$ in (2) allows us to take local structures of each projected source or target-domain instance into consideration, while class labels are implicitly embedded in $\mathbf{M}$.

It is worth noting that, while matching cross-domain marginal distributions in (1) can be viewed as eliminating the domain/dataset bias (as TCA does), matching cross-domain data distributions based on the observed label and sub-domain information (i.e., minimizing (2)) introduces the CCSL the ability to handle imbalanced cross-domain data during adaptation. In Section 4, we will verify the effectiveness of our CCSL for unsupervised domain adaptation with both balanced and unbalanced cross-domain data.

3.3.2 CCSL as TCA or JDA

We note that, both TCA [20] and JDA [17] can be regarded as special cases of our proposed CCSL. For TCA, neither label nor latent domain information are considered when matching cross-domain data distributions. Thus, disregarding (2) would simplify our CCSL as TCA. On the other hand, JDA views each cross-domain pair equally important, if the target-domain instance of this data pair is predicted as the same category as the corresponding source-domain instance is. In other words, if we simply let $m_{ij}^{ST} = 1$ if $y_i^S = \hat{y}_j^T$ without identifying latent domains for adaptation, our propose formulation of (2) would turn into JDA.

3.4. Optimization

To solve the minimization of (1), we first rewrite (1) into the following form:

$$\mathcal{M}_{\phi,d}\left(\mathcal{P}_S(\mathbf{X}_S, \mathbf{y}_S), \mathcal{P}_T(X_T, \mathbf{y}_T)\right) = \mathrm{tr}\left(\mathbf{K}_\phi \mathbf{L}\right), \quad (3)$$

where $\mathbf{K}_\phi \equiv \phi\left(\mathbf{X}\right)^\top \phi\left(\mathbf{X}\right)$ is the kernel matrix constructed over cross-domain data. The matrix $\mathbf{L}$ in (3) is derived as:

$$\mathbf{L} = \mathbf{v}_0 \mathbf{v}_0^\top + \sum_{i,j} m_{ij} \mathbf{v}_{ij} \mathbf{v}_{ij}^\top,$$

$$\text{where} \quad \mathbf{v}_0 = \left[\frac{\mathbf{e}_{N_S}^\top}{N_S}, -\frac{\mathbf{e}_{N_T}^\top}{N_T}\right]^\top$$

$$\mathbf{v}_{ij} = \left[\frac{\left(\mathbf{m}_i^{SS}\right)^\top}{\left\|\mathbf{m}_i^{SS}\right\|_1}, -\frac{\left(\mathbf{m}_j^{TT}\right)^\top}{\left\|\mathbf{m}_j^{TT}\right\|_1}\right]^\top.$$

Note that $\mathbf{e}_N$ is a N dimensional vector of ones. $\mathbf{m}_i^{SS}$ and $\mathbf{m}_j^{TT}$ represent the i and jth column vectors of $\mathbf{M}^{SS}$ and $\mathbf{M}^{TT}$, respectively.

As pointed out in [20], it is computationally expensive to solve the optimization problem of (3). Therefore, following [20, 17], we utilize the *Empirical Kernel Mapping* [22] and predefine a kernel matrix $\mathbf{K} = (\mathbf{K}\mathbf{K}^{-1/2})(\mathbf{K}^{-1/2}\mathbf{K})$. Next, we determine projections $\tilde{\mathbf{A}}$ and $\mathbf{A}$ (both of size $(N_S + N_T) \times k$) for deriving a lower k dimensional space in terms of $\mathbf{K}$. This is achieved by having $\mathbf{K}_\phi = (\mathbf{K}\mathbf{K}^{-1/2}\tilde{\mathbf{A}})(\tilde{\mathbf{A}}^\top \mathbf{K}^{-1/2}\mathbf{K}) = \mathbf{K}\mathbf{A}\mathbf{A}^\top\mathbf{K}$, where $\mathbf{A} = \mathbf{K}^{-1/2}\tilde{\mathbf{A}}$, where $\tilde{\mathbf{A}}$ is to transform the corresponding feature vectors to a lower k-dimensional space.

Algorithm 1 CCSL: Closest Common Space Learning

Input: Data matrix $\mathbf{K}$, source-domain label $\mathbf{y}_S$, dim. k, and α

1. Initialize Initialize: $\mathbf{M} \longleftarrow \mathbf{e}_N \mathbf{e}_N^\top$

while not converged **do**

2. $\mathbf{A} \longleftarrow$ solution of (4)
3. Data embedding $\mathbf{Z} = [\mathbf{Z}_S, \mathbf{Z}_T] \longleftarrow \mathbf{A}^\top \mathbf{K}$
4. Train classifier $f \longleftarrow \{\mathbf{Z}_S, \mathbf{y}_S\}$ and $\hat{\mathbf{y}}_T \longleftarrow f(\mathbf{Z}_T)$
5. Train linear discriminators $\mathbf{W}$
6. $\mathbf{M} \longleftarrow \sigma\left(\mathbf{W}^\top \mathbf{Z}\right)$

end while

Output: Target-domain label $\hat{\mathbf{y}}_T$

Finally, by rewriting $\mathbf{K}_\phi$ in (3), we solve the following objective function for CCSL:

$$\begin{aligned} &\min_{\mathbf{A}} \operatorname{tr}\left(\mathbf{A}^\top \mathbf{K} \mathbf{L} \mathbf{K}^\top \mathbf{A}\right) + \alpha \|\mathbf{A}\|_F^2 \\ &\text{s.t.} \quad \mathbf{A}^\top \mathbf{K} \mathbf{H} \mathbf{K}^\top \mathbf{A} = \mathbf{I}, \end{aligned} \tag{4}$$

where α controls the regularization of $\mathbf{A}$, and $\mathbf{H} = \mathbf{I} - \mathbf{e}_N \mathbf{e}_N^\top / N$ is the centering matrix which preserves data variance after the projection.

By applying $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \cdots, \lambda_k) \in \mathbb{R}^{k \times k}$ as the Lagrange multiplier, solving (4) is equivalent to minimizing the following function:

$$\begin{aligned} \mathcal{L} = &\operatorname{tr}\left(\mathbf{A}^\top \left(\mathbf{K} \mathbf{L} \mathbf{K}^\top + \alpha \mathbf{I}\right) \mathbf{A}\right) \\ &+ \operatorname{tr}\left(\left(\mathbf{I} - \mathbf{A}^\top \mathbf{K} \mathbf{H} \mathbf{K}^\top \mathbf{A}\right) \mathbf{\Lambda}\right). \end{aligned} \tag{5}$$

By setting $\partial \mathcal{L} / \partial \mathbf{A} = \mathbf{0}$, the above problem turns into a generalized eigen-decomposition task. In other words, we calculate the k smallest eigenvectors of the following problem for determining the optimal $\mathbf{A}$:

$$\left(\mathbf{K} \mathbf{L} \mathbf{K}^\top + \alpha \mathbf{I}\right) \mathbf{A} = \mathbf{K} \mathbf{H} \mathbf{K}^\top \mathbf{A} \mathbf{\Lambda}.$$

As summarized in Algorithm 1, we apply the technique of iterative optimization to calculate the projection $\mathbf{A}$, linear discriminators $\mathbf{w}$, and similarity matrix $\mathbf{M}$ for CCSL. Once the closest common space is derived, one can perform classification using projected cross-domain data accordingly.

4. Experiments

4.1. Datasets and Settings

4.1.1 Cross-domain datasets for visual classification

In our experiments, we evaluate the recognition performance of our proposed method on several cross-domain visual classification tasks. We first consider two handwritten digit datasets of MNIST [16] and USPS [14] (denoted as M and U, respectively). The former contains a training set of $60,000$ images of 10 digits, and $10,000$ images are available for testing. The resolution of each image is of size 28×28 pixels. As for USPS, there are 7291 and 2007 images available for training and testing, respectively. Each image in this dataset is of size 16×16 pixels.

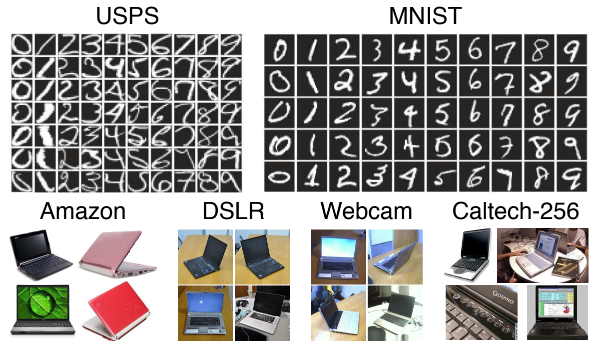


Figure 2: Example images of different datasets for cross-domain visual classification.

We also consider cross-domain object recognition, using the datasets of Caltech-256 (C) [10] and Office [21] datasets. The former consists of real-world object images of 256 categories with at least 80 instances per category, while the latter contains 31 object categories from three different domains, i.e., Amazon (A), DSLR (D), and webcam (W). As suggested by [6, 17], 10 overlapping categories across the above four domains are selected for experiments. Example images of the above datasets are shown in Figure 2.

For fair comparisons, we follow the setting of [20] and randomly sample 2000 and 1800 images from MNIST and USPS (scaled to the same 16×16 pixels), respectively. And, we use pixel intensities are the associated image features. As for cross-domain object recognition, DeCAF_6 features [4] with 4096 dimensions are adopted, since the use of such deep-learning based features have shown very promising results for visual classification [4].

4.1.2 Settings and parameters

To compare our CCSL with existing unsupervised domain adaptation approaches, we consider the methods of *Transfer Component Analysis* (TCA) [20], *Joint Distribution Adaptation* (JDA) [17], and *Transfer Joint Matching* (TJM) [18] in our experiments. We also apply standard SVM trained by source-domain data, which indicates direct recognition without adaptation (denoted as SVM). Although the recent approach of [7] is able to handle mixed source-domain data, its focus is to identify the best subset of the source-domain data, followed by using GFK [8, 6] for performing adaptation. Moreover, the label numbers are assumed to be the same across domains in [7].

It is worth noting that, since no labeled data can be observed in the target domain, performing cross-validation for parameter selection is not applicable. Thus, we sim-

Table 1: Accuracy (%) for cross-domain hand written digit and object classification with balanced cross-domain data. Note that CCSL performs comparably as JDA and TJM do (* indicates cross-domain object recognition only).

S → T	SVM	TCA	JDA	TJM	CCSL
M → U	44.28	52.33	51.78	60.83	53.78
U → M	39.30	46.90	57.80	47.50	58.10
C → A	91.54	90.92	90.92	89.77	93.32
D → A	87.06	88.62	90.28	89.46	90.92
W → A	75.78	80.27	87.02	86.12	89.98
A → C	85.13	82.37	86.33	79.43	87.18
D → C	79.07	79.52	83.88	78.90	84.06
W → C	72.84	74.71	83.64	75.78	82.90
A → D	85.99	87.26	88.54	82.17	87.26
C → D	89.17	89.81	90.36	85.99	87.90
W → D	99.36	100.00	100.00	100.00	96.18
A → W	76.95	74.58	83.78	75.93	83.05
C → W	80.00	78.98	85.08	78.64	82.37
D → W	98.64	99.32	97.98	98.98	96.27
Average*	85.13	85.53	88.98	85.10	88.45

Table 2: Accuracy (%) for cross-domain object recognition with imbalanced label numbers (the best performance is highlighted in bold). Note that the label numbers are 10 and 5 for source and target domain data, respectively.

S → T_5	SVM	TCA	JDA	TJM	CCSL
C → A_5	90.88	90.55	90.20	86.93	**93.32**
D → A_5	85.96	87.90	80.42	83.24	**95.04**
W → A_5	74.10	79.90	76.92	78.91	**91.61**
A → C_5	83.62	80.84	77.49	69.63	**91.88**
D → C_5	77.17	78.45	78.24	65.00	**89.00**
W → C_5	74.09	73.32	75.04	72.63	**83.22**
A → D_5	79.66	80.95	73.69	85.35	**86.55**
C → D_5	**90.66**	87.69	83.05	87.09	85.27
W → D_5	99.24	**100.00**	99.70	**100.00**	99.13
A → W_5	75.12	73.06	71.34	73.45	**85.01**
C → W_5	80.85	71.11	68.85	78.09	**84.10**
D → W_5	98.38	**99.59**	98.16	98.81	96.65
Average	84.14	83.61	81.09	81.59	**90.07**

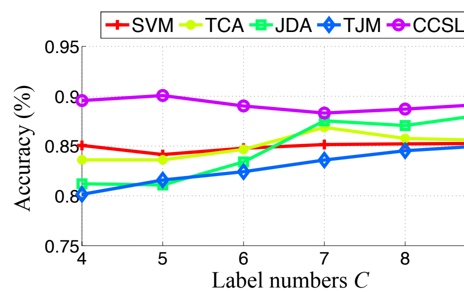


Figure 3: Average results over different target-domain label numbers C for cross-domain object recognition. Note that the source-domain label number is fixed as 10.

ply choose linear SVMs for all approaches (i.e., linear SVMs are trained using projected source-domain data for all MMD-based approaches). For data embedding in TCA, JDA, and CCSL, we apply linear kernels for constructing the kernel matrix as suggested by [17, 20]. As for the remaining parameters, we set the regularization parameter α in (3) as 0.1 and 1 for cross-domain digit and object recognition, respectively. To fix the reduced dimensions for all MMD-based approaches for comparisons, we have $k = 15$ and $k = 100$ for the above two tasks.

4.2. Evaluation

4.2.1 Classification with balanced cross-domain data

For cross-domain handwritten digit recognition, two classification tasks need to be addressed, i.e., M → U and U → M (S → T indicates adapting data from S to T domains). As for cross-domain object recognition, we have a total of 12 cross-domain pairs to be evaluated.

Table 1 lists the recognition results of all methods on the above cross-domain tests. Since all the cross-domain pairs are balanced, i.e., the label and domain numbers across source and target domains are the same, our CCSL produced comparable performance as JDA did. Since TCA and TJM did not utilize any label information during adaptation, degraded performances were obtained.

From Table 1, we see that our CCSL is favorable for target domains with larger sizes (e.g., $|\mathsf{A}| = 958$ and $|\mathsf{C}| = 1123$). This is due to the fact that our CCSL is able to identify proper local data structures for adaptation. Nevertheless, the following experiments using imbalanced cross-domain data will further verify the effectiveness and robustness of our method.

4.2.2 Classification with imbalanced label numbers

For the experiments with imbalanced cross-domain data, we first consider the scenario of imbalanced label numbers across domains. More specifically, we consider the task of cross-domain object recognition, in which the source-domain label number is larger than that in the target domain.

Among the 10 overlapping object categories for Caltech-256 and Office, we randomly select $C = 4 \sim 9$ as the label numbers in the target domain. And, all labeled data of all 10 categories are applied as the source-domain data. Due to space limit, we only present the classification results of $C = 5$ for all domain pairs in Table 2. From this table, we see that TCA, JDA, and TJM were not able to produce satisfactory results, while improved performance was still obtained by our CCSL. The degraded performance of exist-

Table 3: Accuracy (%) for cross-domain object recognition with mixed-domain data. Note that the best performance for each mixed domain pair is highlighted in bold.

S → T	SVM	TCA	JDA	TJM	LM	CCSL
C + D + W → A	91.65	91.34	91.34	89.98	91.75	**93.75**
A + D + W → C	85.66	83.17	86.02	80.50	87.00	**87.98**
D + W → A + C	80.29	80.98	88.66	84.31	86.35	**89.18**
C + W → A + D	93.16	92.96	93.26	91.77	93.45	**93.86**
C + D → A + W	93.44	92.64	92.55	91.40	**93.62**	**93.62**
A + W → C + D	88.44	87.84	89.04	85.62	88.83	**89.38**
A + D → C + W	89.20	87.57	89.67	86.09	89.77	**89.98**
A + C → D + W	87.78	88.75	**91.69**	**91.69**	**91.69**	**91.69**
Average	89.99	89.75	91.58	89.73	90.31	**92.15**

ing MMD-based approaches is due to their assumption of balanced label numbers across source and target domains.

In addition to Table 2, Figure 3 further compares the average performances (over all 12 domain pairs) of different methods using different label numbers C with 10 random trials. From Figure 3, we see that our CCSL performed favorably against existing MMD-based methods, especially when C became smaller. This suggests that the advantage of our CCSL would become clearer if highly imbalanced label numbers are expected to be present across domains.

4.2.3 Classification with mixed-domain data

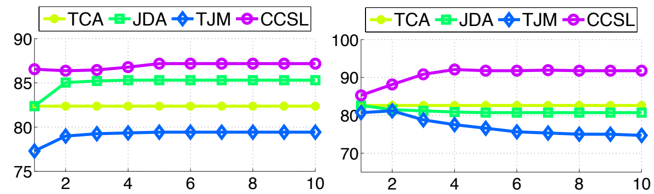
Finally, we consider cross-domain object recognition using mixed-domain data. Table 3 lists and compares the performances of different approaches, including LM [6]. The first two rows in Table 3 represent the scenarios of mixed source-domain data, with unlabeled data to be recognized collected from a single target domain. As for the remaining rows in Table 3, both labeled and unlabeled data are collected from multiple domains, and thus multiple latent domains are expected for both source and target domains.

From Table 3, we observe that improved recognition results were obtained by our CCSL. It can also be seen that, the difference between our CCSL and other recent/baseline approaches was not as significant as those presented in the previous subsection. This implies that, for practical unsupervised domain adaptation task, solving imbalanced label numbers across domains is a more challenging task than that with mixed-domain data. Nevertheless, training data (and their labels) collected in real-world scenarios are typically noisy and imbalanced across domains. As verified above, a robust unsupervised domain adaptation with the ability to handle imbalanced cross-domain data would be preferable.

4.3. Remarks

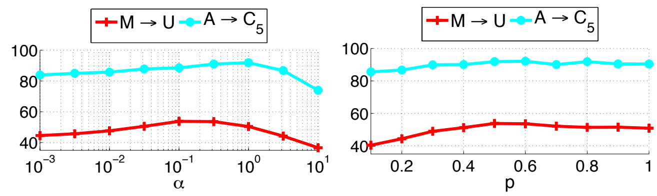
4.3.1 Convergence analysis and parameter sensitivity

We first provide remarks on the convergence issue for our proposed algorithm. For both cross-domain digit and ob-



(a) (b)

Figure 4: Convergence analysis (accuracy (%) vs. number of iterations) on (a) balanced domain pair of A → C and (b) imbalanced domain pair of A → C_5.



(a) (b)

Figure 5: Parameter sensitivity. We show the recognition accuracy (%) over (a) α and (b) p on selected domain pairs.

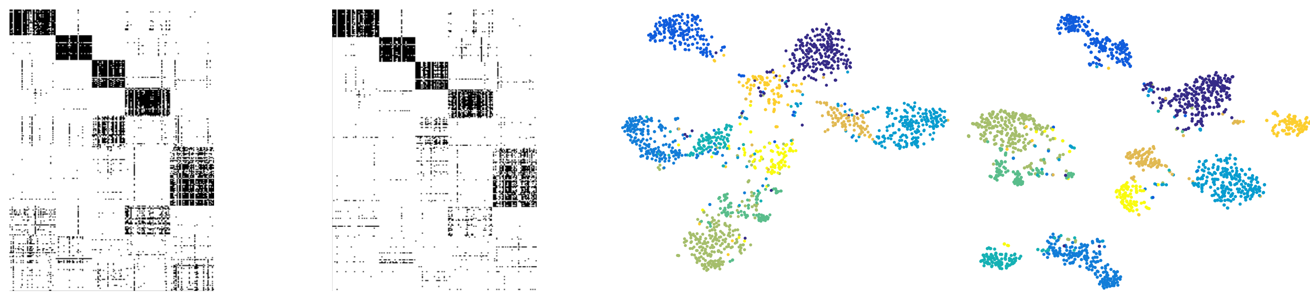
ject recognition, we observe that the optimization of CCSL always converged within 5 iterations for both balanced and imbalanced settings (as shown in Figures 4a and b). We also observe that, when dealing with imbalanced cross-domain data, the convergence of existing MMD-based methods like JDA and TJM does *not* necessarily correspond to non-decreasing performance improvements. Such trends were not observed for the experiments with balanced cross-domain data. This further verifies our advantages in identifying sub-domains for improved adaption.

Figures 5a and b further verify the sensitivity of α in (4) and p in Section 3.3.1. In our experiments, we fix $p = 0.5$ and set $\alpha = 0.1$ and 1 for cross-domain digit and object recognition, respectively. From Figures 5a and b, we see that performance would not be sensitive to the parameters around our choices.

4.3.2 Visualization of adapting imbalanced cross-domain data

In Sections 4.2, we provide experimental results which quantitatively verify the effectiveness of our approach for cross-domain visual classification. To qualitatively support the use of our CCSL for unsupervised domain adaptation (especially for imbalanced cross-domain data), we now discuss the resulting cross-domain data similarity and visualize the data embedding for the adapted data using t-distributed stochastic neighbor embedding (t-SNE) [26].

Figure 6 shows the cross-domain similarity and data embedding analysis for the imbalanced domain pair of A → C_5. To plot the cross-domain similarity, we construct the



(a) (b) (c) (d)

Figure 6: Analysis of cross-domain similarity and data embedding for the imbalanced domain pair of $\mathsf{A} \rightarrow \mathsf{C}_5$. For cross-domain similarity, we show the affinity matrices of cross-domain data derived by (a) JDA and (b) CCSL. For data embedding, we present the 2D visualization of t-SNE for projected cross-domain data derived by (c) JDA and (d) CCSL. In (c) and (d), instances in different colors denote data of different object categories.

affinity matrix, in which each entry denotes the inner product of the associated cross-domain data pair. Once the affinity matrix is obtained, a threshold of 0.8 is applied to binarize this matrix for visualization purposes. Comparing Figures 6(a) and (b), we see a large number of irrelevant entries were nonzero in the affinity matrix of JDA, while the dominant (non-zero) ones in our affinity matrix mainly corresponded to the object categories to be transferred.

Figure 6(c) and (d) illustrate the 2D visualization of t-SNE for adapted cross-domain data (i.e., those projected into the common spaces derived by JDA or CCSL). From these two figures, it is clear that CCSL was able to preserve the label and structural information for cross-domain data with the same class. As for JDA, the separation between projected data of different classes was not sufficient. From the quantitative experiments presented in Sections 4.2, together with the qualitative and visual comparisons provided in this subsection, the effectiveness and robustness of our proposed method can be successfully verified.

5. Conclusion

In this paper, we presented Closest Common Space Learning (CCSL) for unsupervised domain adaptation. In particular, our CCSL is designed to handle mixed-domain data or imbalanced label numbers across domains during adaptation. Solving our proposed algorithm can be viewed as jointly optimizing the tasks of instance reweighting and subspace learning, which exploits label and sub-domain information for data within and across domains. In addition to providing the optimization details for deriving CCSL solutions, we also relate CCSL with popular MMD-based approaches of TCA and JDA. This shows that our CCSL is a robust unsupervised domain adaptation approach for both scenarios of balanced and imbalanced cross-domain data. Finally, we conducted experiments on multiple cross-domain visual classification problems. The empirical results confirmed that our CCSL performs favorably against state-of-the-art unsupervised domain adaptation approaches, especially when imbalanced cross-domain data are presented.

6. Acknowledgement

This work was supported in part by the Ministry of Science and Technology of Taiwan under Grants MOST103-2221-E-001-021-MY2 and MOST104-2221-E-017-016.

References

[1] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann. Unsupervised domain adaptation by domain invariant projection. In *IEEE ICCV*, 2013.

[2] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann. Domain adaptation on the statistical manifold. In *IEEE CVPR*, 2014.

[3] L. Bruzzone and M. Marconcini. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *PAMI*, 2010.

[4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

[5] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *IEEE ICCV*, 2013.

[6] B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *ICML*, 2013.

[7] B. Gong, K. Grauman, and F. Sha. Reshaping visual datasets for domain adaptation. In *NIPS*, 2013.

[8] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. 2012.

[9] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. In *NIPS*, 2006.

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

[11] J. Hoffman, B. Kulis, T. Darrell, and K. Saenko. Discovering latent domains for multisource domain adaptation. In *ECCV*. Springer, 2012.

[12] D.-A. Huang and Y.-C. F. Wang. Coupled dictionary and feature space learning with applications to cross-domain image synthesis and recognition. In *IEEE ICCV*, 2013.

[13] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola. Correcting sample selection bias by unlabeled data. In *NIPS*, 2006.

[14] J. J. Hull. A database for handwritten text recognition research. *PAMI*, 1994.

[15] J. Jiang and C. Zhai. Instance weighting for domain adaptation in NLP. In *ACL*, volume 7, pages 264–271, 2007.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 1998.

[17] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer feature learning with joint distribution adaptation. In *IEEE ICCV*, 2013.

[18] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer joint matching for unsupervised domain adaptation. In *IEEE CVPR*, 2014.

[19] J. Ni, Q. Qiu, and R. Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *IEEE CVPR*, 2013.

[20] S. J. Pan, I. W. Tsang, J. T. Kwok, Q. Yang, et al. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 2011.

[21] K. Saenko et al. Adapting visual category models to new domains. In *ECCV*. 2010.

[22] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 1998.

[23] A. Sharma, A. Kumar, H. Daume, and D. W. Jacobs. Generalized multiview analysis: A discriminative latent space. In *IEEE CVPR*, 2012.

[24] S. Si, D. Tao, and B. Geng. Bregman divergence-based regularization for transfer subspace learning. *IEEE TKDE*, 22, 2010.

[25] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NIPS*, 2008.

[26] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[27] Z. Xu, W. Li, L. Niu, and D. Xu. Exploiting low-rank structure from latent domains for domain generalization. In *ECCV*. Springer, 2014.

[28] Y. Yeh, C. Huang, and Y. Wang. Heterogeneous domain adaptation and classification by exploiting the correlation subspace. 2014.

[29] C. Zhang, Y. Zhang, S. Wang, J. Pang, C. Liang, Q. Huang, and Q. Tian. Undo the codebook bias by linear transformation for visual applications. In *ACM MM*, 2013.

[30] E. Zhong, W. Fan, J. Peng, K. Zhang, J. Ren, D. Turaga, and O. Verscheure. Cross domain distribution adaptation via kernel mapping. In *ACM KDD*, 2009.