Domain-Constraint Transfer Coding for Imbalanced Unsupervised Domain Adaptation

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

Abstract

Unsupervised domain adaptation (UDA) deals with the task that labeled training and unlabeled test data collected from source and target domains, respectively. In this paper, we particularly address the practical and challenging scenario of *imbalanced* cross-domain data. That is, we do not assume the label numbers across domains to be the same, and we also allow the data in each domain to be collected from multiple datasets/subdomains. To solve the above task of *imbalanced domain adaptation*, we propose a novel algorithm of *Domain-constraint Transfer Coding (DcTC)*. Our DcTC is able to exploit latent subdomains within and across data domains, and learns a common feature space for joint adaptation and classification purposes. Without assuming balanced cross-domain data as most existing UDA approaches do, we show that our method performs favorably against state-of-the-art methods on multiple cross-domain visual classification tasks.

1 Introduction

For standard machine learning and pattern recognition problems, training and test data are typically collected from the same domain. This allows the learning model observed by training data to exhibit sufficient generalization, so that the test data can be recognized or fit by the derived model. However, if the above data are collected from different domains, i.e., if the feature distributions of training and test data are very different across domains, one cannot expect the models learned in one domain to be generalized to cope with data in the target domain.

Domain adaptation (DA) deals with the problem in which training and test data are collected from source and target domains, respectively. With the goal to eliminating the domain bias, DA can be divided into two categories depending on the availability of the labeled data in the target domain: *semisupervised* and *unsupervised* domain adaptation. In this paper, we address the task of unsupervised domain adaptation (UDA), which allows labeled data to be presented in the source domain, while only unlabeled data can be observed in the target domain. In particular, we focus on UDA with *imbalanced cross-domain data*. To be more precise, the label numbers across domains can be different, and the source and/or target domain data can be collected from multiple datasets (with unknown numbers).

As we discuss later in Section 2, most existing UDA approaches are not designed to handle imbalanced cross-domain data (Kulis, Saenko, and Darrell 2011; Pan et al. 2011; Gong et al. 2012; Fernando et al. 2013; Baktashmotlagh et al. 2013; Long et al. 2014; Baktashmotlagh et al. 2014). They typically assume that both source and target domains contain data with the same number of categories, and are not particularly designed to deal with mixed-domain problems (i.e., data in each domain are collected from a single dataset or exhibit similar feature distributions). While some recent works like (Hoffman et al. 2012; Gong, Grauman, and Sha 2013b; Xu et al. 2014) have been proposed to address the above mixed-domain problems, they either require the prior knowledge of the domain numbers during the learning process. Nevertheless, existing UDA works typically do not consider the scenario of imbalanced label numbers across domains.

To solve UDA with imbalanced cross-domain data, we propose *Domain-constraint Transfer Coding (DcTC)* in this paper. As detailed in Section 3.2, our DcTC aims to derive a domain invariant space for aligning and representing cross-domain data, while sub-domain locality information can be properly preserved. We perform comprehensive experiments on cross-domain classification with three different settings: standard UDA with balanced cross-domain data, UDA with mixed-domain data, and UDA with imbalanced cross-domain label numbers. Our experimental results would verify the effectiveness of our DcTC in dealing with different cross-domain classification tasks.

2 Related Work

In general, two different strategies have been applied to address the task of unsupervised domain adaptation: *feature space learning* (Pan et al. 2011; Long et al. 2013b; Gong et al. 2012; Baktashmotlagh et al. 2013; Fernando et al. 2013) and *instance reweighting* (Sugiyama et al. 2008; Huang et al. 2006; Chattopadhyay et al. 2013). Some recent works further integrate the above strategies and thus can be viewed as hybrid methods (Gong, Grauman, and Sha 2013a; Long et al. 2014; Baktashmotlagh et al. 2014).

For feature space learning, the general idea is to derive a common feature space for representing projected source and target-domain data, so that the domain mismatch in this space can be diminished. Based on minimizing the maximum mean discrepancy (MMD) for data across domains, transfer component analysis (TCA) (Pan et al. 2011) is proposed to match marginal distributions of cross-domain data under the UDA setting. It is extended by joint distribution adaptation (JDA) (Long et al. 2013b), which additionally minimizes the difference between the class-conditional distributions of cross-domain data. Since no label information is available in the target domain, JDA applies the labels predicted by source-domain classifiers as the pseudo labels for completing the above process.

Different from MMD-based approaches, geodesic flow kernel (GFK) (Gong et al. 2012) projects source and target-domain data onto a Grassmann manifold, while the intermediate samples along the geodesic curve between projected cross-domain data can be applied for representation and adaptation. Subspace alignment (SA) (Fernando et al. 2013) aligns the PCA subspaces derived by source and target-domain data, respectively, so that DA can be performed in the resulting feature space. Transfer sparse coding (TSC) (Long et al. 2013a) applies sparse coding to represent source and target-domain data, while they use data from *both* domains for learning the common domain-invariant dictionary.

On the other hand, the goal of instance reweighting is to select or reweight source-domain instances, so that cross-domain data distributions can be better observed and matched. For example, the work of (Bruzzone and Marconcini 2010) selects source-domain data to learn classifiers for recognizing target-domain instances. The approach of (Chattopadhyay et al. 2013) reweights source-domain instances and actively selects a number of target-domain ones for minimizing domain mismatch. Transfer joint matching (TJM) (Long et al. 2014) observes the relevance of source-domain data to the target-domain ones by introducing a $\ell_{2,1}$-norm regularizer, followed by matching cross-domain distribution. A recent method of (Gong, Grauman, and Sha 2013a) reweights source-domain instances by minimizing the MMD criterion between source and target domain, and apply GFK for completing the adaptation process.

To deal with multiple latent sub-domains for UDA, the approach of (Hoffman et al. 2012) partitions the source domain into a predetermined number of latent sub-domains based on their structural information. Based on the same idea, the method of (Gong, Grauman, and Sha 2013b) applies the partition strategy of maximizing the distinctiveness and learnability of cross-domain data. Extended form exemplar-SVMs (Malisiewicz et al. 2011), low-rank exemplar SVM (LRE-SVM) (Xu et al. 2014) is proposed to deal with UDA with mixed source or target domain data by exploiting their underlying low-rank structures. Nevertheless, the above UDA approaches either require the number of sub-domains as the prior knowledge, or they cannot handle the imbalanced label numbers across domains. In the following section, we will introduce our proposed Domain-constraint Transfer Coding (DcTC), and explain why it is preferable in solving such practical and challenging tasks.

3 Our Proposed Method

3.1 Problem Settings and Motivations

Given the data in source and target domains as $\mathcal{D}_S = \{(\mathbf{x}_1^s, y_1^s), \ldots, (\mathbf{x}_{n_S}^s, y_{n_S}^s)\} = \{\mathbf{X}_S, \mathbf{y}_S\}$ and $\mathcal{D}_T = \{(\mathbf{x}_1^t, y_1^t), \ldots, (\mathbf{x}_{n_T}^t, y_{n_T}^t)\} = \{\mathbf{X}_T, \mathbf{y}_T\}$, respectively, we have $\mathbf{X}_S \in \mathbb{R}^{d \times n_S}$ and $\mathbf{X}_T \in \mathbb{R}^{d \times n_T}$ represent n_S and n_T d-dimensional instances in each domain, and entries in $\mathbf{y}_S \in \mathbb{R}^{n_S \times 1}$ and $\mathbf{y}_T \in \mathbb{R}^{n_T \times 1}$ denote their corresponding labels (from 1 up to C). For unsupervised domain adaptation, $\mathbf{y}_T$ is not known during the training process, and thus our proposed method aims at predicting $\mathbf{y}_T$ using the observed $\mathcal{D}_S$ and $\mathbf{X}_T$. Moreover, when dealing with *imbalanced* cross-domain data, we allow $\mathbf{X}_S$ and $\mathbf{X}_T$ to be collected from multiple datasets (or sub-domains), and the source-domain label number can be greater than or equal to that in the target domain.

Based on approaches like (Pan et al. 2011; Long et al. 2013b; Baktashmotlagh et al. 2013; Long et al. 2014), we share the same goal of learning a domain invariant projection $\mathbf{A} \in \mathbb{R}^{d \times d_c}$, which results in a d_c-dimensional feature space for associating cross-domain data. However, different from the above methods, we do not focus on matching cross-domain data distributions, as existing MMD-based approaches do. This is because such methods typically require the approximation of marginal and/or class-conditional data distributions, which might not be sufficient for matching imbalanced cross-domain data as verified by later experiments.

Inspired by the recent success of sparse coding, we propose Domain-constraint Transfer Coding (DcTC) which can be viewed as a hybrid approach of instance selection and joint feature space learning. When deriving a domain-invariant projection $\mathbf{A} \in \mathbb{R}^{d \times d_c}$ for adaptation, we consider the projected target-domain data $\mathbf{X}_T$ can be linearly and sparsely represented by the source-domain data in that space. In other words, we need to jointly determine $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^{n_T} \in \mathbb{R}^{n_S \times n_T}$, which is the sparse coefficients of the projected target domain data. In addition, inspired by Laplacian sparse coding (Gao et al. 2010) and locality-constrained linear coding (Wang et al. 2010), we further incorporate a data locality constraint for preserving the observed sub-domain structure. As verified by the experiments, by integrating domain-invariant transformation and sparse representation with preserved sub-domain information, our DcTC would exhibit excellent ability in solving UDA problems with imbalanced cross-domain data.

3.2 Domain-constraint Transfer Coding

For standard pattern recognition tasks using sparse representation, one can apply training instances for linearly reconstructing the test data. In other words, if there is no domain difference, UDA can be approached by solving the following problem:

$$\min_{\mathbf{V}} \sum_{i=1}^{n_T} \left\|\mathbf{x}_i^t - \mathbf{X}_S \mathbf{v}_i\right\|_2^2 + \alpha \left\|\mathbf{v}_i\right\|_1, \tag{1}$$

where $\mathbf{v}_i$ is the sparse code of the ith unlabeled test data $\mathbf{x}_i^t$, while α is the parameter for controlling the sparsity.

Unfortunately, for most domain adaptation tasks, the domain difference between source and target domains is not negligible. Without projecting cross-domain data into a properly defined feature space, one cannot apply (1) to describe the target-domain data and predict their labels. Thus, we advance an orthogonal transformation matrix $\mathbf{A}$ as a domain-invariant projection, which allows us to project source and target-domain into a joint feature space for representation and adaptation purposes. Now, we reformulate (1) as follows:

$$\min_{\mathbf{A},\mathbf{V}} \sum_{i=1}^{n_T} \left\| \mathbf{A}^\top \mathbf{x}_i^t - \mathbf{A}^\top \mathbf{X}_S \mathbf{v}_i \right\|_2^2 + \alpha \left\| \mathbf{v}_i \right\|_1 + \lambda \left\| \mathbf{A} \right\|_F^2 \quad \text{s.t. } \mathbf{A}^\top \mathbf{X}\mathbf{H}\mathbf{X}^\top \mathbf{A} = \mathbf{I}. \tag{2}$$

In (2), we have the data matrix $\mathbf{X} = [\mathbf{X}_T, \mathbf{X}_S]$ and $\mathbf{H} = \mathbf{I} - \frac{1}{n_S+n_T}\mathbf{1}$ the centering matrix, where $\mathbf{1}$ is the $(n_S + n_T) \times (n_S + n_T)$ matrix of ones. The parameter λ penalizes the regularizer. As suggested in (Pan et al. 2011; Long et al. 2013b), the constraint for preserving the covariance of the projected cross-domain data is imposed in (2).

There are two major concerns of the above formulation. First, standard sparse representation algorithms like (1) and (2) do not guarantee similar coding outputs given that the input data $\mathbf{x}^t$ are close to each other. More importantly, for UDA problems with imbalanced cross-domain data, one not only needs to deal with possible mismatch of label numbers across domains, there would exist multiple sub-domains within source and/or target domains. This is the reason why we advocate the use of a data locality constraint instead of the sparsity one, aiming at preserving the observed sub-domain structures in the resulting feature space.

More precisely, we replace the ℓ_1-norm term in (2) by a locality term, and solve the following optimization problem:

$$\min_{\mathbf{A},\mathbf{V}} \sum_{i=1}^{n_T} \left\| \mathbf{A}^\top \mathbf{x}_i^t - \mathbf{A}^\top \mathbf{X}_S \mathbf{v}_i \right\|_2^2 + \alpha \left\| \mathbf{d}_i \odot \mathbf{v}_i \right\|_2^2 + \lambda \left\| \mathbf{A} \right\|_F^2 \quad \text{s.t. } \mathbf{A}^\top \mathbf{X}\mathbf{H}\mathbf{X}^\top \mathbf{A} = \mathbf{I}. \tag{3}$$

In (3), the symbol $\odot$ denotes element-wise multiplication, and $\mathbf{d}_i$ is the locality adaptor in which each entry d_{ik} indicates the distance between an instance $\mathbf{x}_i$ and the k_{th} column of codebook in the projected space. We note that, for simplicity, we fix both α and λ as 1 in our work, and we provide additional analysis of parameter sensitivity in our experiments. In our work, we apply an ℓ_2-norm locality adaptor as follows:

$$d_{ik} = \left\| \mathbf{A}^\top \mathbf{x}_i^t - \mathbf{A}^\top \mathbf{x}_k^s \right\|_2. \tag{4}$$

It can be seen that, given labeled source-domain and unlabeled target-domain data, our proposed DcTC learns a domain-invariant transformation $\mathbf{A}$, which exploits the sub-domain structures of cross-domain data (in the resulting feature space) for representation and adaptation. Moreover, since each projected target-domain instance would be sparsely and locally represented by selected (and relevant) source-domain ones, the challenging task of the imbalanced label numbers across domains can be addressed. It is worth repeating that, our approach does not require any prior knowledge on the number of sub-domains as some prior UDA methods do, while most existing UDA algorithms cannot handle the problem of imbalanced label numbers.

After solving (3), we directly apply the transformation matrix $\mathbf{A}$ and the sparse code $\mathbf{v}_i$ of each target-domain instance $\mathbf{x}_i^t$ for performing cross-domain classification. This can be simply achieved by advancing the technique of sparse-representation based classification (SRC) (Wright et al. 2009). That is, the label of $\mathbf{x}_i^t$ is determined by the minimum class-wise reconstruction error:

$$c^* = \arg\min_c \left\| \mathbf{A}^\top \mathbf{x}_i^t - \mathbf{A}^\top \mathbf{X}_S^c \mathbf{v}_i^c \right\|_2^2, \tag{5}$$

where $\mathbf{X}_S^c = [\mathbf{x}_{1,c}^s, \mathbf{x}_{2,c}^s, \cdots, \mathbf{x}_{n_c^s,c}^s] \in \mathbf{R}^{d \times n_c^s}$ denotes the source instances of class c and $\mathbf{v}_i^c$ denotes the sparse code of $\mathbf{x}_i^t$ with respect to class c. Therefore, we have $\mathbf{v}_i = [\mathbf{v}_i^1, \mathbf{v}_i^2, \cdots, \mathbf{v}_i^C]$.

3.3 Optimization

To solve the proposed DcTC of (3), we apply alternative optimization for solving the transformation $\mathbf{A}$ and the sparse code $\mathbf{V}$, as described below.

i) Optimizing transformation matrix A

To solve (3) with respect to $\mathbf{A}$ with fixed $\mathbf{V}$, the first term $\sum_{i=1}^{n_T} \left\| \mathbf{A}^\top \mathbf{x}_i^t - \mathbf{A}^\top \mathbf{X}_S \mathbf{v}_i \right\|_2^2$ can be reformulated as:

$$tr(\mathbf{A}^\top(\mathbf{X}_T - \mathbf{X}_S\mathbf{V})(\mathbf{X}_T - \mathbf{X}_S\mathbf{V})^\top \mathbf{A}) = tr(\mathbf{A}^\top \mathbf{X}\mathbf{W}\mathbf{X}^\top \mathbf{A}),$$

where $\mathbf{X} = [\mathbf{X}_T, \mathbf{X}_S]$ and

$$\mathbf{W} = \begin{bmatrix} \mathbf{I}_{n_T} & -\mathbf{V}^\top \\ -\mathbf{V} & \mathbf{V}\mathbf{V}^\top \end{bmatrix}.$$

Thus, to determine the optimal $\mathbf{A}$, we solve the following problem:

$$\min_{\mathbf{A}} tr(\mathbf{A}^\top \mathbf{X}\mathbf{W}\mathbf{X}^\top \mathbf{A})) + \alpha \sum_{i=1}^{n_T} \left\| \mathbf{d}_i \odot \mathbf{v}_i \right\|_2^2 + \lambda \left\| \mathbf{A} \right\|_F^2 \quad \text{s.t. } \mathbf{A}^\top \mathbf{X}\mathbf{H}\mathbf{X}^\top \mathbf{A} = \mathbf{I}. \tag{6}$$

Note that we apply a specific distance measurement in our locality adaptor $\mathbf{d}_i$, and apply ℓ_2 norm to calculate $\mathbf{d}_i$. As a result, $\sum_{i=1}^{n_T} \left\| \mathbf{d}_i \odot \mathbf{v}_i \right\|_2^2$ in (6) can be rewritten as $\sum_{i=1}^{n_T} \sum_{k=1}^{n_S} d_{ik}^2 \cdot v_{ik}^2 = tr(\mathbf{A}^\top \mathbf{X}\widetilde{\mathbf{W}}\mathbf{X}^\top \mathbf{A})$, in which

$$\widetilde{\mathbf{W}} = \begin{bmatrix} \nu_T & -\nu^\top \\ -\nu & \nu_S \end{bmatrix},$$

$\nu = \mathbf{V} \odot \mathbf{V}$, $\nu_{ik} = \mathbf{v}_{ik}{}^2$, ν_T is a diagonal matrix with $\nu_{Tii} = \sum_{k=1}^{n_S} \nu_{ik}$ (for $i = 1$ to n_T), and ν_S is a diagonal matrix with $\nu_{Skk} = \sum_{i=1}^{n_T} \nu_{ik}$ (for $k = 1$ to n_S).

With the above derivation, we rewrite (6) as follows:

$$\min_{\mathbf{A}} tr(\mathbf{A}^\top \mathbf{X}\mathbf{W}\mathbf{X}^\top \mathbf{A})) + \alpha tr(\mathbf{A}^\top \mathbf{X}\widetilde{\mathbf{W}}\mathbf{X}^\top \mathbf{A})) + \lambda \left\| \mathbf{A} \right\|_F^2 \quad \text{s.t. } \mathbf{A}^\top \mathbf{X}\mathbf{H}\mathbf{X}^\top \mathbf{A} = \mathbf{I}. \tag{7}$$

Algorithm 1 Domain-constraint Transfer Coding (DcTC) for Imbalanced Unsupervised Domain Adaptation

Input: Source and target-domain data $\mathbf{X}_S$, $\mathbf{X}_T$; source-domain label $\mathbf{y}_S$; feature space dimension d_c; regularization terms α and λ

1: Initialize sparse code $\mathbf{V}^0$ by (10) and domain-invariant transformation $\mathbf{A}^0$ by (11)
2: **while** not converge **do**
3: Update $\mathbf{V}$ by (9)
4: Update $\mathbf{A}$ by (8)
5: **end while**
6: Predict the label $\mathbf{y}_T$ of $\mathbf{X}_T$ (5)

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

The Lagrange function of (7) becomes

$$L = tr(\mathbf{A}^\top(\lambda\mathbf{I} + \mathbf{X}(\mathbf{W} + \alpha\widetilde{\mathbf{W}})\mathbf{X}^\top)\mathbf{A}) + (\mathbf{I} - \mathbf{A}^\top\mathbf{X}\mathbf{H}\mathbf{X}^\top\mathbf{A})\Psi,$$

where $\Psi = \mathbf{diag}([\psi_1, \psi_2, \cdots, \psi_{d_c}])$ as the Lagrange multiplier, and d_c is the number of columns in transformation matrix $\mathbf{A}$. Set $\frac{\partial L}{\partial \mathbf{A}} = 0$, $\frac{\partial L}{\partial \Psi} = 0$, we obtain

$$(\lambda\mathbf{I} + \mathbf{X}(\mathbf{W} + \alpha\widetilde{\mathbf{W}})\mathbf{X}^\top)\mathbf{A} = \mathbf{X}\mathbf{H}\mathbf{X}^\top\mathbf{A}\Psi. \quad (8)$$

Finally, the optimal solution $\mathbf{A}$ can be derived by solving the d_c smallest eigenvectors from (8).

ii) Optimizing the sparse code V

To optimize the sparse code $\mathbf{V}$ in (3) given fixed $\mathbf{A}$, we first reformulate the objective function below:

$$\min_{\mathbf{V}} \sum_{i=1}^{n_T} \left\|\mathbf{A}^\top\mathbf{x}_i^t - \mathbf{A}^\top\mathbf{X}_S\mathbf{v}_i\right\|_2^2 + \alpha\left\|\mathbf{d}_i \odot \mathbf{v}_i\right\|_2^2$$
$$= \min_{\mathbf{V}} \sum_{i=1}^{n_T} \left\|\mathbf{A}^\top\mathbf{x}_i^t - \mathbf{A}^\top\mathbf{X}_S\mathbf{v}_i\right\|_2^2 + \alpha\left\|\mathbf{D}_i \cdot \mathbf{v}_i\right\|_2^2,$$

where $\mathbf{D}_i = \mathbf{diag}(\mathbf{d}_i)$. For each sparse coefficient $\mathbf{v}_i$, it can be derived by:

$$\mathbf{v}_i = \left(\mathbf{X}_S{}^\top\mathbf{A}\mathbf{A}^\top\mathbf{X}_S + \alpha\mathbf{D}_i{}^\top\mathbf{D}_i\right)^{-1}\mathbf{X}_S{}^\top\mathbf{A}\mathbf{A}^\top\mathbf{x}_i^t. \quad (9)$$

iii) Remarks on optimization initialization.

Based on the above derivations, we alternate between the optimization of $\mathbf{A}$ and $\mathbf{V}$ for deriving the final solution. However, since the initialization step is critical for performing UDA, we provide additional discussions below.

In our work, we initialize the sparse code $\mathbf{V}$ through

$$\mathbf{V}^0 = \frac{1}{n_S} \cdot \mathbf{1}_{n_S \times n_T}. \quad (10)$$

That is, when starting the determination of the transformation matrix $\mathbf{A}$, we do not enforce our locality constraint and allow the source-domain data to have equal contributions. This would alleviate possible locally optimized solution due to over-emphasizing the contributions of selected source-domain samples in the initial stage of the optimization. Thus, we set $\alpha = 0$ and solve the following eigenvalue decomposition problem in our first iteration:

$$(\lambda\mathbf{I} + \mathbf{X}\mathbf{W}\mathbf{X}^\top)\mathbf{A}^0 = \mathbf{X}\mathbf{H}\mathbf{X}^\top\mathbf{A}^0\Psi. \quad (11)$$

The pseudo code of our proposed DcTC for UDA can be summarized in Algorithm 1.

Figure 1: Example images of *laptop_computer* from the *Office + Caltech-256* datasets.

4 Experiments

4.1 Dataset and Settings

To evaluate the performance of our proposed DcTC for UDA, we choose to address the task of cross-domain visual classification using benchmark datasets with different UDA settings (i.e., balanced/imbalanced label numbers and mixed-domain problems), as described below.

We consider **Office** (Saenko et al. 2010) and **Caltech-256** (Griffin, Holub, and Perona 2007) datasets. The former contains three data domains: *Amazon* (images downloaded from the Internet), *Webcam* (low-resolution images captured by web cameras), and *DSLR* (high-resolution images captured by DSLR cameras). A total of 31 object categories with 4,652 images are available for the Office dataset. As for **Caltech-256**, it consists of 256 object categories with 30,607 images. Following (Gong et al. 2012), 10 overlapping object categories are chosen from the 4 domains of *Amazon* (A), *Webcam* (W), *DSLR* (D), and *Caltech-256* (C) for experiments. Figure 1 shows example images of these 4 domains.

To describe the object images, we apply the DeCAF_6 features (Donahue et al. 2013) with 4096 dimensions, since the use of deep-learning based features have been shown to achieve promising performance for object recognition (Donahue et al. 2013). As for parameter selection, we set $\alpha = \lambda = 1$, and feature dimension $d_c = 100$.

4.2 Evaluation

i) UDA with balanced cross-domain data.

For this standard setting of UDA, we have the same label number (i.e., 10) in both source and target domains, and consider all 12 cross-domain combinations from A, C, D, and W. We compare our DcTC with five popular UDA methods: GFK (Gong et al. 2012), JDA (Long et al. 2013b), SA (Fernando et al. 2013), LM (Gong, Grauman, and Sha 2013a), and TJM (Long et al. 2014). When presenting the performance of the above approaches, we apply their released code with the same DeCAF_6 features for fair comparisons.

Table 1: Performance of imbalanced UDA on Office+Caltech with multiple sub-domains. Note that **S** and **T** denote source and target domains, respectively.

S→T	GFK	GFK (latent)				LRE-SVM	DcTC
		DLD (Match)	DLD (Ensemble)	RVD (Match)	RVD (Ensemble)		
A, C, D → W	79.78	49.05	84.75	70.1	91.86	91.83	**93.32**
A, C, W → D	84.85	77.45	87.26	83.44	89.17	82.91	**98.73**
C, D, W → A	84.91	85.61	91.34	87.57	**92.48**	90.83	92.28
A, D, W → C	79.69	64.41	82.99	79.03	84.15	86.01	**87.62**
D, W → A, C	77.06	80.87	**86.35**	69.57	84.48	84.32	86.11
A, C → D, W	77.73	55.95	87.39	78.34	**90.71**	86.39	85.84
C, W → A, D	84.82	72.07	83.86	81.76	89.87	91.07	**93.36**
C, D → A, W	83.59	46.15	84.68	76.19	87.31	91.35	**93.77**
A, W → C, D	78.48	65.53	84.53	69.95	84.22	86.62	**89.3**
A, D → C, W	78.4	72.05	81.59	76.88	81.17	86.93	**89.14**
average	80.93	66.91	85.47	77.28	87.54	87.83	**90.94**

Table 2: Average performance P_C for balanced UDA on Office+Caltech. Note that DcTC† denotes our DcTC without exploiting the locality adaptor, and DcTC* indicates the use of NN instead of SRC for classification in the derived space.

Method	GFK	JDA	SA	LM
P_C	81.91	87.55	80.94	85.19
Method	TJM	DcTC†	DcTC*	DcTC
P_C	85	83.98	87.39	**88.8**

Table 3: Cross-domain classification on Office+Caltech with imbalanced cross-domain label numbers. Note that the source-domain label number is fixed at 10, while the target-domain label number $C_T \leq 10$.

C_T	GFK	JDA	SA	LM	TJM	DcTC
1	64.74	48.95	44.4	81.16	76.64	**87.11**
2	71.85	81.5	61.01	80.09	83.53	**89.53**
3	74.45	79.9	63.45	79.28	82.87	**89.08**
4	76.52	80.6	67.09	81.05	82.84	**89.62**
5	78.34	83.41	69.96	81	82.38	**89.7**
6	78.7	83.8	71.07	80.73	82.3	**88.64**
7	80.99	85.19	73.31	82.53	82.94	**88.81**
8	80.55	85.1	75.69	83.34	83.2	**88.47**
9	81.31	86.49	78.24	83.81	84.05	**88.49**
10	81.91	87.55	80.94	85.19	85	**88.8**

Table 2 lists and average recognition performance of different approaches. We note that, due to space limit, we only present the average recognition rates over all 12 cross-domain pairs. From this table, we see that our achieved promising performance on the standard UDA problems with balanced cross-domain data. It is worth noting that, under the standard balanced UDA settings, achieved comparable or slightly improved performance over existing UDA approaches would be sufficiently satisfactory.

We note that, in Table 2, we also consider the removal of the locality adaptor (i.e., $\alpha = 0$ in (3)), and observed degraded performance (denoted as DcTC†). This confirms that the introduction of this term would be crucial for UDA. In addition, to further verify the effectiveness of our derived feature space for UDA, we also consider the use of nearest neighbor (NN) classifiers for recognizing projected target instance (instead of SRC) (denoted as DcTC*). We see that the use of NN achieved comparable result as SRC did, and thus we confirm that the feature space derived by our DcTC exhibits excellent adaptation ability in associating cross-domain data.

ii) Imbalanced UDA with multiple sub-domains

For the second part of the experiments, we consider the imbalanced UDA setting with multiple sub-domains observed in source and target domains. For comparison purposes, we consider recent UDA approaches of Discovering Latent Domains (DLD) (Hoffman et al. 2012), Reshaping Visual Datasets (RVD) (Gong, Grauman, and Sha 2013b), and LRE-SVM (Xu et al. 2014), which are all proposed to handle such scenarios.

Table 1 lists and compares the classification results of different approaches on different imbalanced cross-domain combinations. For DLD and RVD, we follow (Gong, Grauman, and Sha 2013b; Xu et al. 2014) and apply GFK (Gong et al. 2012) for adapting the identified sub-domains for classification. Note that, two different fusing strategies of *ensemble* and *match* are considered in (Gong, Grauman, and Sha 2013b). The former reweights the classifiers trained from each sub-domains based on the observed sub-domain probabilities, while the later identifies the most relevant sub-domain via MMD for classification.

From the results presented in Table 1, we see that our DcTC outperformed existing UDA methods, which share the same goal of coping with mixed-domain problems. This confirms the use of our DcTC for jointly learning of feature space and data representation, with additional capability in exploiting and associating sub-domain structures for improved cross-domain classification.

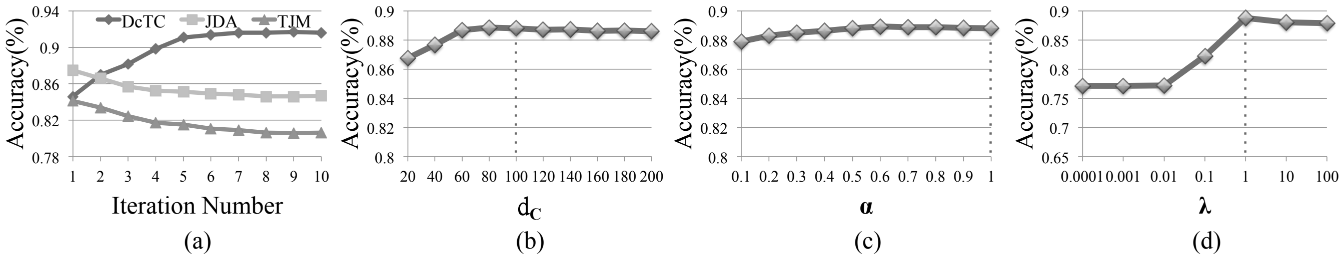

Figure 2: Convergence analysis and parameter sensitivity tests. The parameters of interest are (a) iteration number (for the cross-domain pair of $A \to C_5$), (b) feature dimension d_c, and regularizers (c) α and (d) λ.

iii) UDA with imbalanced cross-domain label numbers Finally, we conduct UDA experiments with imbalanced cross-domain label numbers. More specifically, we consider that the source-domain has labeled data from all 10 object categories, while the unlabeled data (to be classified) in the target domain are from 1 up to 10 categories. Similarly, due to space limit, we only present the average classification rates of all 12 cross-domain pairs from Office+Caltech (with 10 random trials), and lists the performance in Table 3. From the results shown in Table 3, we see that methods of instance reweighting/selection like LM and TJM were less sensitive to the label number changes in the target domain, while approaches based on feature space learning like GFK, JDA, and SA were not able to handle imbalanced label numbers well. Nevertheless, our DcTC performed favorably against the above popular UDA approaches on this challenging setting, without any knowledge of sub-domain or label numbers in either domain. Take the extreme case of the first row in Table 3 for example, in which only data from one object category was presented in the target domain, our DcTC still achieved over a 87% classification rate, while those of others were close to or below 81%.

Based on the above experiments with different UDA settings, it can be seen that DcTC not only achieved comparable results as existing UDA methods on standard/balanced settings, it was particularly favorable for the challenging and practical scenarios of imbalanced UDA. Therefore, the effectiveness and robustness of our proposed method can be successfully verified.

4.3 Convergence analysis and parameter sensitivity

Recall that, our DcTC alternates between the derivation of the domain-invariant transform $\mathbf{A}$ and the sparse code of target-domain data $\mathbf{V}$ for optimization. Since existing methods such as JDA and other MMD-based approaches also apply iterative optimizing processes, we now evaluate the convergence performance among these approaches.

For all the experiments, we observe that DcTC always converged within 7-10 iterations (see Figure 2(a) for example). However, for MMD-based approaches such as JDA and TJM, they did not exhibit similar properties since they were not able to handle imbalanced UDA well. Take Figure 2(a) for example, we consider the case where *Amazon* as the source domain (all 10 classes) and 5 randomly selected categories from *Caltech* as the target domain. We see that, with the increase of the iteration number, the performance of JDA and TJM dropped due to incorrect matching of cross-domain marginal and conditional data distributions.

As for evaluating the parameter sensitivity of our DcTC, we consider the feature dimension d_c, and two regularization parameters α (for the locality adaptor) and λ (for preventing overfitting of $\mathbf{A}$). We conduct additional experiments by varying the values of the above parameters, and show the corresponding averaged performance (with balanced UDA settings) in Figures 2(b) to (d). From these figures, we see that the performance was generally not sensitive to such parameter choices, and thus our default parameter settings would be reasonable for implementation.

5 Conclusion

In our paper, we proposed *Domain-constraint Transfer Coding (DcTC)* for solving imbalanced unsupervised domain adaptation problems. That is, in addition to only the presence of labeled data in the source but not target domain, one would expect imbalanced label and sub-domains numbers across domains. Based on joint feature space learning for cross-domain data representation, our DcTC is able to exploit the latent sub-domains in each domain for improved adaptation and classification. Different from existing UDA approaches, our method does not require the prior knowledge of the sub-domain numbers, nor the assumption of equal label numbers across domains. From our experiments, we confirmed that our DcTC exhibited excellent ability in addressing cross-domain classification tasks with different and challenging UDA settings.

6 Acknowledgements

This work is 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

Baktashmotlagh, M.; Harandi, M. T.; Lovell, B. C.; and Salzmann, M. 2013. Unsupervised domain adaptation by domain invariant projection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, 769–776. IEEE.

Baktashmotlagh, M.; Harandi, M. T.; Lovell, B. C.; and Salzmann, M. 2014. Domain adaptation on the statistical manifold. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2481–2488. IEEE.

Bruzzone, L., and Marconcini, M. 2010. Domain adaptation problems: A dasvm classification technique and a circular validation strategy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32(5):770–787.

Chattopadhyay, R.; Fan, W.; Davidson, I.; Panchanathan, S.; and Ye, J. 2013. Joint transfer and batch-mode active learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 253–261.

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

Fernando, B.; Habrard, A.; Sebban, M.; and Tuytelaars, T. 2013. Unsupervised visual domain adaptation using subspace alignment. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, 2960–2967. IEEE.

Gao, S.; Tsang, I. W.-H.; Chia, L.-T.; and Zhao, P. 2010. Local features are not lonely–laplacian sparse coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 3555–3561. IEEE.

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

Gong, B.; Grauman, K.; and Sha, F. 2013a. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *Proceedings of The 30th International Conference on Machine Learning*, 222–230.

Gong, B.; Grauman, K.; and Sha, F. 2013b. Reshaping visual datasets for domain adaptation. In *Advances in Neural Information Processing Systems*, 1286–1294.

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

Hoffman, J.; Kulis, B.; Darrell, T.; and Saenko, K. 2012. Discovering latent domains for multisource domain adaptation. In *Computer Vision–ECCV 2012*. Springer. 702–715.

Huang, J.; Gretton, A.; Borgwardt, K. M.; Schölkopf, B.; and Smola, A. J. 2006. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, 601–608.

Kulis, B.; Saenko, K.; and Darrell, T. 2011. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 1785–1792. IEEE.

Long, M.; Ding, G.; Wang, J.; Sun, J.; Guo, Y.; and Yu, P. S. 2013a. Transfer sparse coding for robust image representation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 407–414. IEEE.

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

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 (CVPR), 2014 IEEE Conference on*, 1410–1417. IEEE.

Malisiewicz, T.; Gupta, A.; Efros, A.; et al. 2011. Ensemble of exemplar-svms for object detection and beyond. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 89–96. IEEE.

Pan, S. J.; Tsang, I. W.; Kwok, J. T.; and Yang, Q. 2011. Domain adaptation via transfer component analysis. *Neural Networks, IEEE Transactions on* 22(2):199–210.

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

Sugiyama, M.; Nakajima, S.; Kashima, H.; Buenau, P. V.; and Kawanabe, M. 2008. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information processing systems*, 1433–1440.

Wang, J.; Yang, J.; Yu, K.; Lv, F.; Huang, T.; and Gong, Y. 2010. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 3360–3367. IEEE.

Wright, J.; Yang, A. Y.; Ganesh, A.; Sastry, S. S.; and Ma, Y. 2009. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31(2):210–227.

Xu, Z.; Li, W.; Niu, L.; and Xu, D. 2014. Exploiting low-rank structure from latent domains for domain generalization. In *Computer Vision–ECCV 2014*. Springer. 628–643.