Bridging the Domain Shift by Domain Adaptive Dictionary Learning

Hongyu Xu
hyxu@umiacs.umd.edu
Jingjing Zheng
zjngjng@umiacs.umd.edu
Rama Chellappa
rama@umiacs.umd.edu

Department of Electrical and Computer Engineering and the Center for Automatic Research, UMIACS
University of Maryland
College Park, Maryland, USA

Abstract

Domain adaptation (DA) tackles the problem where data from the training set (source domain) and test set (target domain) have different underlying distributions. In this paper, we propose a novel domain-adaptive dictionary learning framework to generate a set of intermediate domains. These intermediate domains form a smooth path and bridge the gap between the source and target domains. Specifically, we not only learn a common dictionary to encode the domain-shared features, but also learn a set of domain-specific dictionaries to model the domain shift. The separation of the common and domain-specific dictionaries enables us to learn more compact and reconstructive dictionaries for domain adaptation. These dictionaries are learned by alternating between domain-adaptive sparse coding and dictionary updating steps. Meanwhile, our approach gradually recovers the feature representations of both source and target data along the domain path. By aligning all the recovered domain data, we derive the final domain-adaptive features for recognition. Extensive experiments on cross-domain face and object recognition show that our approach significantly outperforms state-of-the-art methods.

1 Introduction

In real world scenarios, the assumption that the training data (***source*** domain) and test data (***target*** domain) are sampled from the same distribution is often challenged. For instance, training and testing images may be acquired under different environments, viewpoints and illumination conditions in application such as face recognition [2, 8], object recognition [13, 14, 15, 26], human detection [31] and video concept detection [10, 11, 32]. Recently, many works have been proposed to adapt the classifier trained using the source domain data to perform well on target samples [5, 6, 9, 12, 20, 22, 23, 25, 28, 32]. This is known as the ***domain adaptation*** (DA) problem. In this paper, we focus on the more challenging unsupervised DA problem where the samples in the target domain are unlabeled. Moreover, it would be highly desirable for recognition systems to automatically adapt to a different domain without any additional labeling effort.

Recently, the most promising approaches for the unsupervised DA problem focus on developing intermediate feature representations [13, 15, 22, 29] along a virtual path connecting the source and target domains. [15] generated intermediate subspaces by sampling

Pages 96.1-96.12
DOI: https://dx.doi.org/10.5244/C.29.96

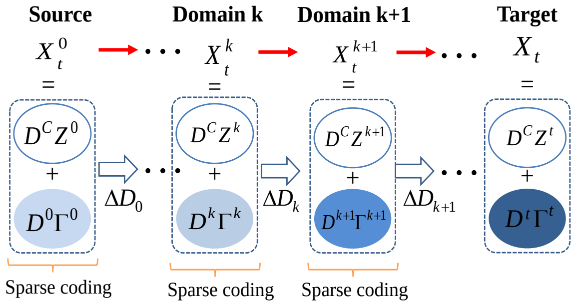


Figure 1: **Our dictionary learning framework.** The overall learning process consists of three steps: (1) Dictionary learning in source and target domains. At the beginning, we first learn the common dictionary D^C, domain-specific dictionaries D^0 and D^t for source and target domains. (2) Domain-adaptive sparse coding. At the k-th step, we enforce the recovered feature representations of target data in all available domains to have the same sparse codes, while adapting the newest obtained dictionary D^k to better represent the target domain. Then we multiply dictionaries in the k-th domain with the corresponding sparse codes to recover feature representations of target data X_t^k in this domain. (3) Dictionary updating. We update D^k to find the next domain-specific dictionary D^{k+1} by further minimizing the reconstruction error in representing the target data. Then we alternate between sparse coding and dictionary updating steps until the stopping criteria is satisfied.

the geodesic path connecting the source and target subspaces on the Grassmann manifold. Instead of sampling a few intermediate subspaces as in [15], [13] integrated an infinite number of intermediate subspaces to derive a geodesic flow kernel to model the domain shift. However, the subspaces obtained using principal component analysis (PCA) in both methods may not well represent the original data and some useful information for adaptation may be lost. In order to overcome the limitation of PCA subspaces, a recent work [22] used a dictionary to represent each domain, as non-orthogonal atoms (columns) in the dictionary provide more flexibility to model and adapt the domain data.

In this paper, we propose a novel domain-adaptive dictionary learning approach to generate a set of intermediate domains which bridge the gap between source and target domains. Our approach defines two types of dictionaries: a common dictionary and a domain-specific dictionary. The common dictionary shared by all domains is used to extract domain-shared features, whereas the domain-specific dictionary which is incoherent to the common dictionary models the domain shift. The separation of the common dictionary from domain-specific dictionary enables us to learn more compact and reconstructive dictionaries for deriving domain-adaptive features. All these dictionaries are learned using the procedure illustrated in Figure 1. First, we learn a common dictionary D^C by minimizing the reconstruction error of both source and target data. Then combined with the common dictionary, we learn a set of domain-specific dictionaries by alternating between the following two steps: 1) domain-adaptive sparse coding: we learn domain-adaptive sparse codes Γ and Z by enforcing the feature representations of the target data to have the same sparse codes in all available domains. 2) dictionary updating: we update the current domain-specific dictionary to generate the next domain-specific dictionary such that the reconstruction error of target data is further minimized. This step not only guarantees that the next domain-specific dictionary will better represent the target data, but also ensures that the intermediate domains gradually adapt to the target domain. Finally, we apply domain-adaptive sparse codes combined with domain dictionaries to construct the final domain-adaptive features for recognition.

Ni et al.'s work in [22] may be the closest to our work in spirit. However, our approach differs in the following three aspects: (1) The separation of the common dictionary from domain-specific dictionaries. We aim to learn ***both*** the common dictionary and domain-specific dictionaries to represent each intermediate domain while [22] used only a ***single*** dictionary to represent each domain. Our approach has two advantages over [22]. First, our approach can better represent the domain data because the reconstruction error of domain data obtained using our method is smaller as demonstrated by Figure 2(c) in Section 4. Second, the domain-specific dictionaries can better model the domain changes because the domain-shared features are accounted for separately. (2) The regularization of sparse coding. In each step, we regularize the representation of the target data along the path to have the same sparse codes, which are further used for dictionary updating in the next step. However, the sparse codes used in [22] for dictionary updating are only adaptive between the ***neighboring*** domains. Therefore, the sparse representations of target data in [22] are ***not*** domain-adaptive, while the sparse representations in our approach are ***domain-adaptive***. Moreover, the intermediate domains generated by our approach are smoother and incorporate the domain change in a better way, which will be verified and discussed in section 4.2. (3) The construction of final features. We use the domain-adaptive sparse codes across all the domains multiplied by the dictionaries to represent source and target data, while [22] only uses the sparse code decomposed with source and target dictionaries respectively to represent the new features. Therefore, compared to [22], our approach generates more robust and domain-adaptive features. We make the following contributions:

- We learn a common dictionary to extract features shared by all the domains and a set of domain-specific dictionaries to encode the domain shift. The separation of the common dictionary from domain-specific dictionaries enables us to learn more compact and reconstructive representations for learning.
- We propose a new formulation to incrementally adapt the dictionaries learned from the source domain to reduce the reconstruction error of target data.
- We recover the feature representations of source and target data in all intermediate domains and extract novel domain-adaptive features by concatenating these intermediate features.
- We present empirical results for the tasks of object recognition and face recognition across pose, illumination, and blur variations, that are better than state-of-the-art algorithms.

2 Related Work

Recently, dictionary-based approaches [22, 25, 27] have been proposed for unsupervised DA. [25] learned a parametric modeled dictionary by aligning dictionaries from both domains. [27] jointly learned the projections of data in two domains, and a latent dictionary which can represent both domains in the projected low-dimensional space. [22] generated a set of intermediate domains and dictionaries which smoothly adapt the source domain to the target domain.

Another fruitful line of work is the subspace-based approaches [12, 13, 15, 28, 29]. [13, 15] created the intermediate domain subspaces along the geodesic on the Grassmann manifold connecting the source and target domains. [28] proposed to jointly learn domain-adaptive features and the classifiers on the target domain using an information-theoretic measure. [29] proposed an approach based on the parallel transport to incrementally learning the intermediate domains. [5, 14, 19, 20] attempted to match the distributions of the source and target samples by domain sample re-weighting and feature matching.

Semi-supervised DA methods mainly focus on using samples with labels in the target domain to reduce the differences in data distribution [9, 14, 19, 23, 25, 26, 32]. Transformation-based methods [18, 26] learn linear or nonlinear transformations such that samples of the same class from different domains become closer. Classifier-based methods [3, 4, 17, 32] adapt the Support Vector Machine (SVM) trained in the source domain to correctly classify labeled target samples. A survey on the visual domain adaptation could be found in [24].

3 Domain Adaptive Dictionary Learning

Let $X_s \in \mathbb{R}^{d\times N_s}$, $X_t \in \mathbb{R}^{d\times N_t}$ be the feature representations of source and target data respectively, where d is the feature dimension, N_s and N_t are the number of samples in the two domains. The feature representations of recovered source and target data in the k-th intermediate domain are denoted as $X_s^k \in \mathbb{R}^{d\times N_s}$ and $X_t^k \in \mathbb{R}^{d\times N_t}$ respectively. The common dictionary is denoted as D^C, whereas source-specific and target-specific dictionaries are denoted as D^0, D^t respectively. Similarly, we use $D^k, k = 1...N$ to denote the domain-specific dictionary for the k-th domain, where N is the number of intermediate domains. We set all the dictionaries to be of the same size $\in \mathbb{R}^{d\times n}$.

Our objective is to learn the common dictionary and a set of domain-specific dictionaries for generating intermediate domains. Starting from D^0 in the source domain, we sequentially learn the intermediate domain-specific dictionaries $\{D^k\}_{k=1}^N$ to gradually reduce the reconstruction error of the target data. Our domain-adaptive dictionary learning approach (**DADL**) consists of three steps: (1) Dictionary initialization. At the beginning, we first learn the common dictionary D^C and two domain-specific dictionaries D^0, D^t for the source and target domains respectively. (2) Domain-adaptive sparse coding. At the k-th step, we learn domain-adaptive sparse codes of target data and recover the feature representations of target data in the k-th domain. (3) Dictionary updating. We update the current domain-specific dictionary D^k to find the next domain-specific dictionary D^{k+1} by further minimizing the residual error in representing the target data. We alternate between dictionary updating and sparse coding steps until the stopping criteria is satisfied.

3.1 Dictionary Learning in Source and Target Domains

At the beginning, we learn the common dictionary D^C, source-specific dictionaries D^0 and target-specific dictionary D^t. Given source and target data X_s and X_t, we solve for D^C by minimizing the reconstruction error of both source and target data as follows:

$$\min_{D^C,Z^0,Z^t} ||X_s - D^C Z^0||_F^2 + ||X_t - D^C Z^t||_F^2 \quad s.t.\ \forall i, \quad ||z_i^0||_0 \le T, ||z_i^t||_0 \le T \tag{1}$$

where $Z^0 = [z_1^0 ... z_{N_s}^0] \in \mathbb{R}^{n\times N_s}, Z^t = [z_1^t ... z_{N_t}^t] \in \mathbb{R}^{n\times N_t}$ are sparse representations of X_s and X_t respectively, T specifies the sparsity that each sample has fewer than T dictionary atoms (columns) in its decomposition.

Given the learned D^C and corresponding sparse codes Z^0 and Z^t, we learn domain-specific dictionaries D^0 and D^t by further reducing the reconstruction error of the source and target data. The objective function for learning D^0 and D^t is given as follows:

$$\min_{D^0,\Gamma^0} \|X_s - D^C Z^0 - D^0\Gamma^0\|_F^2 + \lambda\|D^0 D^{C^T}\|_F^2 \;\; s.t. \; \forall i, \;\; \|z_i^0\|_0 + \|\alpha_i^0\|_0 \le T \tag{2}$$
$$\min_{D^t,\Gamma^t} \|X_t - D^C Z^t - D^t\Gamma^t\|_F^2 + \lambda\|D^t D^{C^T}\|_F^2 \;\; s.t. \; \forall i, \;\; \|z_i^t\|_0 + \|\alpha_i^t\|_0 \le T \tag{3}$$

where $\Gamma^0 = [\alpha_1^0 ... \alpha_{N_s}^0] \in \mathbb{R}^{n\times N_s}$ and $\Gamma^t = [\alpha_1^t ... \alpha_{N_t}^t] \in \mathbb{R}^{n\times N_t}$ are sparse representations of X_s and X_t with respect to D^0 and D^t, and λ is the regularization parameter. The first term in both (2) and (3) is the reconstruction error of domain data using both the common dictionary and corresponding domain-specific dictionary. The second term is the inner product of the atoms from different dictionaries, which encourages D^C to be incoherent to the domain-specific dictionaries. This incoherence term minimizes the correlation between D^C and $\{D^0, D^t\}$, thus it enables our approach to exploit domain-shared features and domain changes separately. We describe the optimization of the objective functions (2) and (3) in supplementary materials.

3.2 Domain-adaptive Sparse Coding

At the k-th step, assume we have already generated (k-1) intermediate domains and domain-specific dictionaries denoted as $\{X_t^i\}_{i=1}^{k-1}$ and $\{D^i\}_{i=1}^{k-1}$ respectively. Now given a newly obtained domain-specific dictionary D^k for the k-th domain, we want to obtain sparse representations of target data X_t in the k-th domain. In order to achieve this goal, we not only reconstruct X_t using dictionaries from the k-th domain, but also reconstruct the recovered target data X_t^i in each intermediate domain using dictionaries from that domain. Moreover, we regularize the sparse representation of X_s, X_t and X_t^i to be the same. This regularization step ensures that the sparse representations of target data across all available domains are the same (i.e. ***domain-adaptive***). We solve for domain-adaptive sparse codes across all the available domains as follows:

$$\begin{aligned} Z^k, \Gamma^k = & \arg\min_{Z,\Gamma} \|X_t - D^C Z - D^k\Gamma\|_F^2 + \sum_{i=0}^{k-1} \|X_t^i - D^C Z - D^i\Gamma\|_F^2 + \|X_t - D^C Z - D^t\Gamma\|_F^2 \\ & s.t. \quad \forall i, \|z_i\|_0 + \|\alpha_i\|_0 \le T \end{aligned} \tag{4}$$

where $Z^k = [z_1^k ... z_{N_t}^k], \Gamma^k = [\alpha_1^k ... \alpha_{N_t}^k]$ are the solved sparse representations of target data in the k-th domain, $X_t^i = D^C Z^i + D^i \Gamma^i$ are the recovered feature representations of target data in the i-th domain obtained in previous iteration steps. The objective function in (4) has two terms:

1. The first term is the reconstruction error of target data when encoded using dictionaries from the k-th domain. This term is called ***domain shifting*** term, because it adapts dictionaries in the k-th domain to better represent the target data.

2. The second term in (4) sums the reconstruction errors of recovered feature representations of target data in all the intermediate domains. The last term is the reconstruction error of target data in the target domain. These two terms are called ***domain adaptive*** terms. This is because we regularize both X_t and X_t^i to have the same sparse codes. It means that feature representations of recovered target data in different domains will have the same sparse codes when encoded using dictionaries from each domain. This regularization will guarantee that sparse codes are domain-adaptive, such that the domain changes are encoded only in domain-specific dictionaries.

The above objective function (4) could be rewritten as follows:

$$Z^k, \Gamma^k = \arg\min_{Z,\Gamma} \|\tilde{X} - \tilde{D}[\, Z \;\; \Gamma \,]^T\|_F^2 \tag{5}$$

where $\tilde{X} = \left[X_t^T, X_t^T, X_t^{0T}, \ldots, X_t^{k-1T}\right]^T$ and $\tilde{D} = \begin{bmatrix} D^{tT}, D^{kT}, D^{0T}, \ldots, D^{k-1T} \\ D^{cT}, D^{cT}, D^{cT}, \ldots, D^{cT} \end{bmatrix}^T$. We can solve (5) as a LASSO problem to compute the sparse codes as in [21]. Then we recover the feature representations of target data in the k-th domain X_t^k as follows: $X_t^k = D^C Z^k + D^k \Gamma^k$.

3.3 Domain-specific Dictionary Updating

After sparse coding at the k-th step, we will update D^k to find the next domain-specific dictionary D^{k+1} by further reducing the reconstruction error of target data in the k-th domain. Let J^k denote the target reconstruction residue in the k-th domain, which is computed as follows:

$$J^k = X_t - D^C Z^k - D^k \Gamma^k \tag{6}$$

where Z^k and Γ^k are the sparse codes obtained for reconstructing X_t in the k-th step. We further reduce the target reconstruction residue J^k by adjusting D^k by $\Delta D^k \in \mathbb{R}^{d\times n}$, which is solved from the following problem:

$$\min_{\Delta D^k} \|J^k - \Delta D^k \Gamma^k\|_F^2 + \eta \|\Delta D^k\|_F^2 \tag{7}$$

The objective function in (7) has two terms. The first term ensures that the adjustment ΔD^k will further reduce the target reconstruction residue J^k. While the second term penalizes the abrupt changes between two adjacent domain-specific dictionaries so that the intermediate domains smoothly adapt to the target domain. The parameter η controls the balance between these two terms. Since the problem in (7) is a ridge regression problem, we solve for ΔD^k by setting the first derivative to be zeros as in [22] and obtain:

$$\Delta D^k = J^k \Gamma^{kT} (\eta I + \Gamma^k \Gamma^{kT})^{-1} \tag{8}$$

where $I \in \mathbb{R}^{n\times n}$ is the identity matrix. The next domain-specific dictionary D^{k+1} is obtained as: $D^{k+1} = D^k + \Delta D^k$. In addition, we normalize each column in D^{k+1} to be a unit vector.

Proposition 1. *The residue J^k in (6) is non-increasing with respect to D^C, D^k, ΔD^k and corresponding sparse codes Z^k, Γ^k, i.e. $\|J^k - \Delta D^k \Gamma^k\|_F^2 \le \|J^k\|_F^2$.*

The non-increasing property of the residue J^k ensures that the source-specific dictionary D^0 gradually adapts to the target-specific dictionary D^t through a set of intermediate domain-specific dictionaries D^k. The proof is given in the supplementary material.

After the domain-specific dictionary update, we increase k by 1, and alternate between the sparse coding step in section 3.2 and the dictionary updating step in section 3.3 until the stopping criteria is reached. We summarize our approach in Algorithm 1.

3.4 Derivation of New Features for Domain Data

Until now we have obtained the common dictionary D^C, domain-specific dictionaries $D^k, k \in [0, N]$. The transition path made up of D^c and the set of domain-specific dictionaries D^k models the domain shift. We will make use of it to derive new domain-adaptive representations for source and target data.

Algorithm 1 Our DADL framework

1: **Input:** source data X_s , target data X_t, sparsity level T, parameter λ, η, stopping threshold δ
2: **Output:** D^C, D^0 and D^t
3: compute D^C using (1)
4: compute D^0, D^t by solving the objective function in (2) and (3).
5: $k = 0$
6: **while** stopping criteria is not reached **do**
7: compute domain-adaptive sparse codes Z^k, Γ^k using equation (4)
8: compute the reconstruction error J^k using equation (6).
9: compute the adjustment ΔD^k using equation (8)
10: $D^{k+1} \leftarrow D^k + \Delta D^k$
11: normalize D^{k+1} to have unit atoms.
12: $X_t^{k+1} \leftarrow D^C Z^k + D^k \Gamma^k$
13: $k \leftarrow k+1$
14: Check the stopping criteria $\|\Delta D^k\|_F \leq \delta$
15: **end while**
16: **Final Output:** D^C, $D^k, k \in [0,N]$ and D_t.

Since the recovered feature representations of target data $X_t^k, k \in [0,N]$ in all intermediate domains are already available, we first recover feature representations of source data $X_s^k, k \in [0,N]$ in each intermediate domain. We iteratively recover X_s^k in a similar way as X_t^k. The only difference is that all the dictionaries are already learned and fixed during the learning of X_s^k. Specifically, at the k-th iterative step, we obtain the sparse representations of source data that are adaptive across all domains by solving the following problem:

$$Z_s^k, \Gamma_s^k = arg \min_{Z,\Gamma} \|X_s - D^C Z - D^t \Gamma\|_F^2 + \sum_{i=1}^{k-1} \|X_s^i - D^C Z - D^i \Gamma\|_F^2 \quad s.t. \forall i, \quad \|z_i\|_0 + \|\alpha_i\|_0 \leq T \tag{9}$$

where $Z_s^k = [z_{s_1}^k \ldots z_{s_{N_s}}^k], \Gamma_s^k = [\alpha_{s_1}^k \ldots \alpha_{s_{N_s}}^k]$ are sparse representations of source data in the k-th domain, $X_s^i = D^C Z_s^i + D^i \Gamma_s^i$ are recovered feature representations of source data in the i-th domain obtained in previous iteration steps. The objective function in (9) consists of two terms. The first term is the reconstruction error of source data using dictionaries from the target domain while the second term is the sum of reconstruction error of recovered feature representations of source data in all intermediate domains. Similarly, we enforce both X_s^0 and X_s^i to have the same sparse codes. After sparse coding in the k-th step, we recover the feature representations of source data in the k-th domain as follows: $X_s^k = D^C Z_s^k + D^k \Gamma_s^k$.

We use the sparse codes obtained in the last iterative step to derive the new feature representations for the source and target data. The new augmented feature representation of source and target data are $\tilde{X}_s = [\tilde{X}_s^0, \ldots, \tilde{X}_s^N]$ and $\tilde{X}_t = [\tilde{X}_t^0, \ldots, \tilde{X}_t^N]$ respectively, where $\tilde{X}_s^i = D^C Z_s^N + D^i \Gamma_s^N$ and $\tilde{X}_t^i = D^C Z_t^N + D^i \Gamma_t^N$ and Z_s^N, Z_t^N, Γ_s^N, Γ_t^N are the sparse codes obtained in the last iterative step where $k = N$. The final stage of recognition across all the domains is performed using an SVM classifier trained on new feature vectors after dimension reduction via the Principal Component Analysis (PCA).

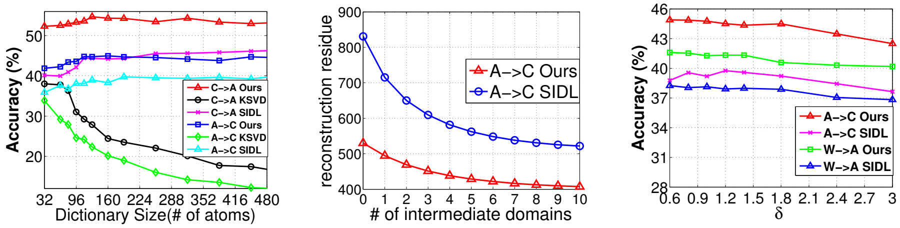
4 Experiments

4.1 Object Recognition

We evaluate our methods for cross-domain object recognition on the benchmark office dataset introduced in [26]. We selected 2533 images from 10 object classes common to four different domains *i.e.* Caltech, Amazon, DSLR, Webcam as in [13] for our experiments. Image

Methods	C→A	C→D	C→W	A→C	A→W	A→D	W→C	W→A	W→D	D→C	D→A	D→W
K-SVD [1]	38.0	19.8	21.3	33.9	23.5	22.3	17.1	16.7	46.5	22.6	14.3	46.8
GFK [13]	40.4	41.1	40.7	37.9	35.7	36.3	29.3	35.5	85.9	30.3	36.1	79.1
SA [12]	39.0	39.6	23.9	35.3	38.6	38.8	32.3	37.4	77.8	38.9	38.0	83.6
SIDL [22]	43.3	42.3	36.3	40.4	37.9	33.3	36.3	38.3	86.2	36.1	39.1	86.2
TJM [20]	46.7	44.6	38.9	39.4	42.0	45.2	30.2	30.0	89.2	31.4	32.8	85.4
DIP [5]	50.0	49.0	47.6	43.3	46.7	42.8	37.0	42.5	86.4	39.0	40.5	86.7
SIE [6]	51.9	52.5	47.3	44.5	**48.6**	43.2	39.9	**44.1**	89.3	38.9	39.1	88.6
Ours	**54.7**	**53.7**	**48.1**	**45.3**	44.5	**45.8**	**40.1**	41.8	**93.6**	**39.3**	**41.7**	**92.4**

Table 1: **Object classification accuracies of different approaches on the benchmark dataset [26].**



(a) dictionary size (b) average reconstrcution error (c) stopping threshold δ

Figure 2: **The effects of dictionary size, average reconstruction error and stopping threshold δ on office datasets [26].**

representation is based on SURF [7] features similar to those in [13, 26]. Specifically, all the images were firstly resized to have the same width and converted to grayscale. Second, the SURF detector [7] was used to extract local scale-invariant interest points. Then a random subset of these interest point descriptors was selected and quantized to 800 visual words by k-means clustering. Each image was represented by a 800-dimensional histogram.

Following the protocol in [26], we selected 8 labeled images per category when Webcam, DSLR and Caltech are used as source domains, and 20 labeled images when Amazon is the source domain. We ran 20 different trials corresponding to different selections of labeled source data and report the average recognition accuracy in Table 1. It is seen that our method achieves the best performance for a majority of combinations of source and target domains. In particular, our method consistently outperforms SIDL [22] which is most similar to ours. This is because [22] only regularizes two adjacent domains to have the identical pairwise sparse codes and the learned dictionaries do not fully capture the domain changes. However, our method encodes the domain changes in the domain-specific dictionaries by encouraging feature representation of different domain data to have the same domain-adaptive sparse codes.

In order to evaluate the effect of dictionary size on our approach, we choose two different combinations of source and target domains and plot the results in Figure 2(a). Our approach gains significant improvement over K-SVD [1] since we bridge the domain shift by generating intermediate domains. Our approach also outperforms SIDL [22] by a large margin of 4.5%. This is because we learn more compact and reconstructive dictionaries to represent target data, which leads to much lower reconstruction errors, as demonstrated in Figure 2(b). The dictionary size is set to be 128 or 256 based on the source sample size in all the experiments. We also evaluate our approach with varying values of stopping threshold δ as shown in Figure 2(c). It can be seen that both [22] and the proposed approach converge in fewer steps with increasing value of δ, thus generates fewer number of intermediate domains. In addition, our approach is insensitive to the regulization parameter η, which is chosen from 1500 to 2500 throughout all the experiments. The final dimensionality after PCA is between

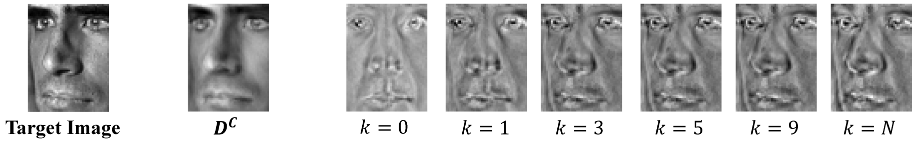


Figure 3: **Recovered face images of a target face image along the intermediate domains.** The first image is the original target face image, the second image is the component of the recovered face image corresponding to the common dictionary. The remaining six images are the components of the recovered face images corresponding to domain-specific dictionaries.

(a) Pose Variation

	c11	c29	c05	c37	average
Ours	**86.7**	**98.5**	95.6	**89.7**	**92.6**
[22]	76.5	**98.5**	**98.5**	88.2	90.4
[20]	83.8	**98.5**	95.6	82.4	90.1
[13]	63.2	92.7	92.7	76.5	81.3
[16]	78.0	91.0	93.0	89.0	87.8
[1]	48.5	76.5	80.9	57.4	65.8

(b) Gaussian Blur Kernels

σ	2	5	7
Ours	**88.9**	**82.7**	**80.5**
[22]	84.0	78.2	76.5
[20]	67.4	64.4	63.8
[13]	81.1	75.9	72.1
[2]	69.1	61.6	55.3
[8]	72.4	24.8	17.3
[1]	49.1	34.6	29.2

(c) Motion Blur Kernels

L	3	7	13
Ours	**97.9**	**89.7**	**77.4**
[22]	95.6	86.5	75.7
[20]	71.8	69.4	60.0
[13]	91.3	84.9	70.7
[2]	81.8	77.4	54.5
[8]	82.3	70.7	35.1
[1]	85.0	56.5	25.9

Table 2: **Recognition accuracies of different approaches for the CMU-PIE dataset [30] across pose variation, Gaussian blur kernels and motion blur kernels respectively.** Each column in (a) corresponds to a non-frontal pose. The columns in (b) and (c) correspond to Gaussian kernels with different values of the standard deviation σ or motion blur kernels with different values of length L.

60 and 140.

4.2 Face Recognition across Pose and Blurs

We evaluate our methods for face recognition across pose and blur variation on CMU-PIE dataset [30]. This dataset is a controlled face dataset of 68 subjects with a total of $41,368$ images. Each subject has 13 images under 9 different poses, 21 different illuminations and 4 different expressions. We choose 5 different poses of face images ranging from frontal to $\pm 45^o$. The four non-frontal poses are denoted as *c05* (yaw about -22.5^o), *c29* (yaw about 22.5^o), *c11* (yaw about 45^o) and *c37* (yaw about -45^o).

4.2.1 Benefits of the Common Dictionary and Domain-specific Dictionaries

We first demonstrate the benefits of the separation of the common dictionary from domain-specific dictionaries. Specifically, we selected frontal face images as the source domain and face images from pose *c11* (yaw about 45^o) as the target domain. We chose a face image from the target domain and recovered feature representations of this face image in intermediate domains using the proposed method. Since the recovered face images have two components corresponding to the common dictionary and domain-specific dictionaries respectively, we visualize the two components of recovered face images separately in Figure 3. It can be seen that the components corresponding to domain-specific dictionaries in intermediate domains gradually adapt from the frontal face to non-frontal faces. This demonstrates that the domain-specific dictionaries have the ability to encode the domain shift due to different yaw angles.

4.2.2 Face Recognition across Poses and Blurs

The second experiment we carried out is face recognition across pose variation. We selected the front-illuminated face images to be the labeled source domain. Face images with the same illumination condition under four different non-frontal poses formed the different target domains. The task is to classify the unlabeled face images from the target domain. As shown in Table 2(a), our method outperforms the other methods except the case where the target pose is $c05$ in [22]. It is interesting to note that when the pose variations are large, [16] which relies on a generic training set to build pose model has higher average recognition accuracies than the unsupervised DA in [13]. However, our method demonstrates improved performances over both [16] and other domain adaptation approaches [13, 20, 22] when pose variations are large.

We also evaluated our approach for face recognition across blur and illuminations. For a fair comparison, we followed the protocol presented in [22] to construct source and target domains. Specifically, we choose 34 subjects under first 11 illumination conditions to compose the source domain. The target domain was formed by the remaining images acquired under 10 other illumination conditions. We synthesized the domain shift by applying two different types of blur kernels to the target data: 1) Gaussian blur kernel with different standard deviations from 2 to 7, and 2) motion blur kernel with different lengths from 3 to 13 along $\Theta = 135^o$. In summary, the domain shift consist of two components. The first is a change in illumination direction and the second component is due to blur.

Tables 2(b) and (c) show the accuracies of different methods for face recognition across Gaussian blur and motion blur respectively. The proposed DADL method consistently achieves the best performance. In addition, since both illumination and blur variations exist in the domain shift, LPQ [2] which is only blur robust and albedo [8] which is only illumination insensitive are not able to handle all the domain changes. Moreover, our method outperforms [22], which demonstrates the benefits of learning both common and domain-specific dictionaries.

5 Conclusion

We presented a novel domain adaptive dictionary learning framework for unsupervised domain adaptation. We first learned a common dictionary to recover features shared by all domains. Then we acquired a set of domain-specific dictionaries, which generates a transition path from source to target domains. The common dictionary is essential for reconstruction while domain-specific dictionaries are able to bridge the domain shift. Final feature representations are recovered by utilizing both common and domain-specific dictionaries. We extensively evaluated our approach on two benchmark datasets and the experimental results clearly confirmed the effectiveness of our approach.

6 Acknowledgement

This work was supported by cooperative agreement FA8750-13-2-0279 from the Defense Advanced Research Projects Agency.

References

[1] M Aharon, M Elad, and A Bruckstein. K-SVD : An algorithm for designing of overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[2] T. Ahonen, E. Rahtu, V. Ojansivu, and J. Heikkilä. Recognition of blurred faces using local phase quantization. In *ICPR*, pages 1–4, 2008.

[3] Lorenzo Torresani Alessandro Bergamo. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In *NIPS*, 2010.

[4] Yusuf Aytar and Andrew Zisserman. Tabula rasa: Model transfer for object category detection. In *ICCV*, pages 2252–2259, 2011.

[5] Mahsa Baktashmotlagh, Mehrtash Tafazzoli Harandi, Brian C. Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *ICCV*, pages 769–776, 2013.

[6] Mahsa Baktashmotlagh, Mehrtash Tafazzoli Harandi, Brian C. Lovell, and Mathieu Salzmann. Domain adaptation on the statistical manifold. In *CVPR*, pages 2481–2488, 2014.

[7] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008.

[8] Soma Biswas, Gaurav Aggarwal, and Rama Chellappa. Robust estimation of albedo for illumination-invariant matching and shape recovery. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):884–899, May 2009.

[9] Hal Daumé III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, June 2007.

[10] Lixin Duan, Ivor W. Tsang, Dong Xu, and Stephen J. Maybank. Domain transfer svm for video concept detection. In *CVPR*, pages 1375–1381, 2009.

[11] Lixin Duan, Dong Xu, Ivor W. Tsang, and Jiebo Luo. Visual event recognition in videos by learning from web data. In *CVPR*, pages 1959–1966, 2010.

[12] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2013.

[13] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073, 2012.

[14] Boqing Gong, Kristen Grauman, and Fei Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *ICML*, volume 28, pages 222–230, 2013.

[15] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, pages 999–1006, 2011.

[16] Ralph Gross, Iain Matthews, and Simon Baker. Appearance-based face recognition and light-fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(4):449–465, April 2004.

[17] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A. Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *ECCV Part I*, pages 158–171, 2012.

[18] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, pages 1785–1792, 2011.

[19] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S. Yu. Transfer feature learning with joint distribution adaptation. In *ICCV*, pages 2200–2207, 2013.

[20] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S. Yu. Transfer joint matching for unsupervised domain adaptation. In *CVPR*, pages 1410–1417, 2014.

[21] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, January 2010.

[22] Jie Ni, Qiang Qiu, and Rama Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *CVPR*, pages 692–699, 2013.

[23] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. In *IJCAI*, pages 1187–1192, 2009.

[24] Vishal M. Patel, Raghuraman Gopalan, and Rama Chellappa. Visual domain adaptation: An overview of recent advances. *IEEE Signal Processing Magazine*, July 2014.

[25] Qiang Qiu, Vishal Patel, Pavan Turage, and Rama Chellappa. Domain adaptive dictionary learning. In *ECCV*, pages 631–645, 2012.

[26] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226, 2010.

[27] Sumit Shekhar, Vishal M. Patel, Hien Van Nguyen, and Rama Chellappa. Generalized domain-adaptive dictionaries. In *CVPR*, pages 361–368, 2013.

[28] Yuan Shi and Fei Sha. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. In *ICML*, 2012.

[29] Ashish Shrivastava, Sumit Shekhar, and Vishal M. Patel. Unsupervised domain adaptation using parallel transport on grassmann manifold. In *WACV, 2014*, pages 277–284, 2014.

[30] Terence Sim, Simon Baker, and Maan Bsat. The cmu pose, illumination, and expression database. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(12):1615–1618, 2003.

[31] David Vázquez, Antonio Manuel López, Javier Marín, Daniel Ponsa, and David Gerónimo Gomez. Virtual and real world adaptation for pedestrian detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(4):797–809, 2014.

[32] Jun Yang, Rong Yan, and Alexander G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *ACM Multimedia*, pages 188–197, 2007.