Coupled Projections for Semi-supervised Adaptation of Dictionaries

Sumit Shekhar, *Student Member, IEEE,* Vishal M. Patel, *Member, IEEE,*, Hien V. Nguyen, *Member, IEEE,*, and Rama Chellappa, *Fellow, IEEE*

***Abstract*—Data-driven dictionaries have produced state-of-the-art results in various classification tasks. However, when the target data has a different distribution than the source data, the learned sparse representation may not be optimal. In this paper, we investigate if it is possible to optimally represent both source and target by a common dictionary. Specifically, we describe a technique which jointly learns projections of data in the two domains, and a latent dictionary which can succinctly represent both the domains in the projected low-dimensional space. The algorithm is modified to learn a common discriminative dictionary, which can be further used for classification. The algorithm can be use for adaptation across multiple domains and is extensible to non-linear feature space. The proposed approach does not require any explicit correspondence between the source and target domains, and shows good results even when there are only a few labels available in the target domain. Further, it can also be used for heterogenous domain adaptation, where different features are extracted for different domains. Various recognition experiments show that the method performs on par or better than competitive state-of-the-art methods.**

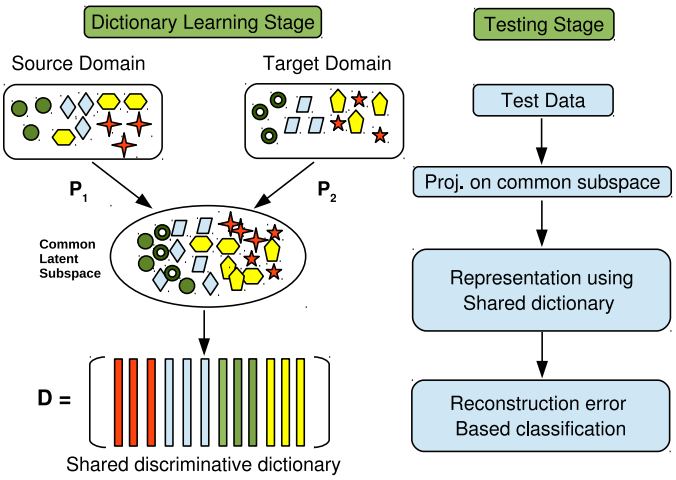


Fig. 1. Overview of the proposed dictionary learning method.

I. Introduction

The study of sparse representation of signals and images has attracted tremendous interest in last few years. Sparse representations of signals and images require learning an over-complete set of bases called a dictionary along with linear decomposition of signals and images as a combination of few atoms from the learned dictionary. Olshausen and Field [30] in their seminal work introduced the idea of learning dictionary from data instead of using off-the-shelf bases. Since then, data-driven dictionaries have been shown to work well for both image restoration [11] and classification tasks [46].

The efficiency of dictionaries in these wide range of applications can be attributed to the robust discriminant representations that they provide by adapting to the particular data samples. However, the learned dictionary may not be optimal if the target data has different distribution than the data used for training. These variations are commonplace in vision problems, and can happen due to changes in image sensor (web-cams vs SLRs), camera viewpoint, illumination conditions, etc. It has been shown that such changes can cause significant degradation in classifier performance [7]. Adapting dictionaries to new domains is a challenging task, but has hardly been explored in the vision literature. Yangqing *et al.* [22] considered a special case where corresponding samples from each domain were available, and learned a dictionary for each domain. More recently, Qiu *et al.* [33] proposed a method for adapting dictionaries for smoothly varying domains using regression. However, in practical applications, target domains are scarcely labeled, and domain shifts may result in abrupt feature changes (e.g., changes in resolution when comparing web-cams to DSLRs). Moreover, high dimensional features are often extracted for object recognition. Hence learning a separate dictionary for each domain will have a severe space constraint, rendering it unfeasible for many practical applications. A subspace interpolation based method was proposed for adapting dictionaries in [29]. However, this method cannot be used for heterogenous domain adaptation, where different features are extracted for different domains.

In view of the above challenges, we propose a robust method for learning a single dictionary to optimally represent both source and target data. As the features may not be correlated well in the original space, we project data from both the domains onto a common low-dimensional space, while maintaining the manifold structure of data. Simultaneously, we learn a compact dictionary which represents projected data from both the domains well. As the final objective is classification, we learn a class-wise discriminative dictionary. This joint optimization method offers several advantages in

Sumit Shekhar is with the Department of Electrical and Computer Engineering and the Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742 (e-mail: sshekha@umiacs.umd.edu).

Vishal M. Patel is with the Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742 (e-mail: pvishalm@umd.edu).

Hien V. Nguyen is with the Siemens Corporate Research, Princeton, NJ 08540 (e-mail: hien@umiacs.umd.edu).

Rama Chellappa is with the Department of Electrical and Computer Engineering and the Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742 (e-mail: rama@umiacs.umd.edu).

terms of generalizability and efficiency of the method. Firstly, learning separate projection matrix for each domain makes it easy to handle any changes in feature dimension and type in different domains. It also makes the algorithm conveniently extensible to handle multiple domains. Further, learning the dictionary on a low-dimensional space makes the algorithm faster, and irrelevant information in original features is discarded. Moreover, joint learning of dictionary and projections ensures that the common internal structure of data in both the domains is extracted, which can be represented well by sparse linear combinations of dictionary atoms.

An additional contribution of the paper is an efficient optimization technique to solve this problem. Using kernel methods, the proposed algorithm can be easily made non-linear, and the resulting optimization problem has a few simple update steps. Further we extensively evaluate the method for different recognition scenarios and show that the proposed method is comparable with other recent algorithms for domain adaptation. We also demonstrate that the algorithm converges quickly and is efficient.

A. Paper Organization

The paper is organized in six sections. In Section II, we describe some of the related works. The algorithm is formulated in Section III, and the extension to non-linear case is described in Section IV. The classification scheme for the learned dictionary is described in Section V. Experimental results are presented in Section VI, and the final concluding remarks are made in VII.

II. Related Work

In this section, we survey the recent domain adaptation works and the related sparse coding literature.

A. Domain Adaptation

The problem of adapting classifiers to new visual domains has recently gained importance in the vision community. Several approaches have been proposed for this problem, which can be broadly categorized into following categories:

1) Feature transform-based approaches: The idea of domain adaptation in vision community was introduced by Saenko *et al.* [35], which learnt a symmetric transformation between domains represented by same features. This was extended to general domain shifts in Kulis *et al.* [24] by learning an asymmetric transformation between domains. In [21], a transformation of source data onto target space is learnt, such that the joint representation is low-rank. Further, Baktashmotlagh *et al.* [2] proposed learning feature transformation for kernel mean matching between domains for adaptation. A subspace alignment-based method was also explored in [12].

2) Manifold interpolation-based approaches: Gopalan *et al* introduced the idea of interpolation between subspaces of different domains on Grassmann manifold [16]. This was extended to learning a kernel distance between domains in [15]. A class-wise adaptation scheme based on parallel transport on manifold was introduced in [42].

3) Classifier transform-based approaches: Many methods have been proposed to adapt classifiers between domains for adaptation. A method for adapting SVMs across domains was proposed for concept detection in [48]. Similar methods based on transforming SVMs have been proposed in [9], [10]. A multiple kernel learning-based approach for domain adaptation was proposed in [8]. Recently, a method for adaptation by reconstructing target classifiers using source classifiers was explored in [51].

4) Other approaches: A feature augmentation method was proposed in [25]. Gong *et al.* [14] described a method of choosing landmarks in the target domain for adaptation. An information theortic clustering-based adaptation approach was proposed in [41]. Recently, deep learning has also been used for domain adaptation [6], [5].

B. Sparse Coding

Here, we review some of the related works in sparse coding literature. Han *et al.* [20] suggested learning a shared embedding for different domains, along with a sparsity constraint on the representation. However, they assume pre-learned projections, which may not be optimal. In the dictionary learning literature, Yang *et al.* [47] and Wang *et al.* [44] proposed learning dictionary pairs for cross-modal synthesis. Similarly, methods for joint dimensionality reduction and sparse representation have also been proposed [50], [13], [26], [28]. Additional methods may be found within these references. A preliminary version of this paper [39] discussed projection-based approach for adaptation of sparse dictionaries.

III. Problem Framework

The classical dictionary learning approach minimizes the representation error of the given set of data samples subject to a sparsity constraint [1]. Let $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N] \in \mathbb{R}^{d \times N}$ be the data matrix. Then, the K-atoms dictionary, $\mathbf{D} \in \mathbb{R}^{d \times K}$, can be trained by solving the following optimization problem

$$\{\mathbf{D}^*, \mathbf{X}^*\} = \arg\min_{\mathbf{D},\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \text{ s.t. } \|\mathbf{x}_i\|_0 \leq T_0 \ \forall i$$

where, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \in \mathbb{R}^{K \times N}$ is the sparse representation of $\mathbf{Y}$ over $\mathbf{D}$, and T_0 is the sparsity level. Here, $\|.\|_0$-norm counts the number of nonzero elements in a vector and $\|.\|_F$ is the Frobenius norm of a matrix.

Now, consider a special case, where we have data from two domains, $\mathbf{Y_1} \in \mathbb{R}^{d_1 \times N_1}$ and $\mathbf{Y_2} \in \mathbb{R}^{d_2 \times N_2}$. We wish to learn a shared K-atoms dictionary, $\mathbf{D} \in \mathbb{R}^{d_f \times K}$ and mappings $\mathbf{P}_1 \in \mathbb{R}^{d_f \times d_1}$, $\mathbf{P}_2 \in \mathbb{R}^{d_f \times d_2}$ onto a common low-dimensional space, which will minimize the representation error in the projected space. Formally, we desire to minimize the following cost function:

$$\mathcal{C}_1(\mathbf{D}, \mathbf{P_1}, \mathbf{P_2}, \mathbf{X_1}, \mathbf{X_2}) = \|\mathbf{P_1}\mathbf{Y_1} - \mathbf{D}\mathbf{X_1}\|_F^2 + \|\mathbf{P_2}\mathbf{Y_2} - \mathbf{D}\mathbf{X_2}\|_F^2 \quad (1)$$

subject to sparsity constraints on $\mathbf{X_1}$ and $\mathbf{X_2}$. However, minimizing $\mathcal{C}_1(\mathbf{D}, \mathbf{P_1}, \mathbf{P_2}, \mathbf{X_1}, \mathbf{X_2})$ will result in trivial solution as $\mathbf{P_i}$s can be set to $\mathbf{0}$. To overcome this, we regularize the solution space to get meaningful solutions.

A. Regularization

It will be desirable if the projections, while bringing the data from two domains to a shared subspace, do not lose too much information available in the original domains. To facilitate this, we add a PCA-like regularization term which preserves energy in the original signal, given as:

$$\mathcal{C}_2(\mathbf{P_1}, \mathbf{P_2}) = \|\mathbf{Y_1} - \mathbf{P_1^T P_1 Y_1}\|_F^2 + \|\mathbf{Y_2} - \mathbf{P_2^T P_2 Y_2}\|_F^2$$
$$\text{s.t. } \mathbf{P_i P_i^T} = \mathbf{I},\ i = 1, 2 \tag{2}$$

It is easy to show after some algebraic manipulations that the costs $\mathcal{C}_1$ and $\mathcal{C}_2$, after ignoring the constant terms in $\mathbf{Y}$, can be written as:

$$\mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}) = \|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}\|_F^2, \tag{3}$$
$$\mathcal{C}_2(\tilde{\mathbf{P}}) = -\text{trace}((\tilde{\mathbf{P}}\tilde{\mathbf{Y}})(\tilde{\mathbf{P}}\tilde{\mathbf{Y}})^T) \tag{4}$$

where,

$$\tilde{\mathbf{P}} = [\mathbf{P_1}\ \mathbf{P_2}],\ \tilde{\mathbf{Y}} = \begin{pmatrix} \mathbf{Y_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y_2} \end{pmatrix}, \text{and } \tilde{\mathbf{X}} = [\mathbf{X_1}\ \mathbf{X_2}].$$

Thus, the form of $\mathcal{C}_2$ is similar to trace minimization problem [23]. Thus, the regularization can be generalized to different dimensionality reduction techniques. We describe some of the possible methods below:

1) **Manifold preserving regularization:** Let $\mathbf{W_1} \in \mathbb{R}^{N_1 \times N_1}$ and $\mathbf{W_2} \in \mathbb{R}^{N_2 \times N_2}$ be affinity matrices calculated from $\mathbf{Y_1}$ and $\mathbf{Y_2}$ using different methods in literature [36], [3]. The manifold preserving mapping can then be formulated as:

$$\mathcal{C}_2(\tilde{\mathbf{P}}) = -\sum_{i=1}^{2} \text{trace}(\mathbf{P_i Y_i})(\mathbf{I} - \mathbf{W_i})(\mathbf{I} - \mathbf{W_i^T})(\mathbf{P_i Y_i})^T$$
$$\text{s.t. } \mathbf{P_i P_i^T} = \mathbf{I},\ i = 1, 2$$

Other possible manifold-based regularizations can also be explored in [23].

2) **Discriminative regularization:** Let $\mathbf{H_{i,j}} = \mathbf{1}_{n_{i,j}} \mathbf{1}_{n_{i,j}}^T$ $i = 1, 2, j = 1, \cdots, C$ where, C is the number of classes in data and $n_{i,j}$ is the number of samples in class j for domain i and $\mathbf{1}_{n_{i,j}}$ is a column vector of length $n_{i,j}$. Define

$$\mathbf{H_i} = \text{diag}[\mathbf{H_{i,1}}, \cdots, \mathbf{H_{i,C}}].$$

Then, discriminative LDA-like regularization can be formulated as [23]:

$$\mathcal{C}_2(\tilde{\mathbf{P}}) = -\sum_{i=1}^{2} \text{trace}(\mathbf{P_i Y_i})(\mathbf{I} - \mathbf{H_i})(\mathbf{P_i Y_i})^T$$
$$\text{s.t. } (\mathbf{P_i Y_i})(\mathbf{P_i Y_i})^\mathbf{T} = \mathbf{I},\ i = 1, 2$$

In this paper, we focus on the PCA-like regularization (4), other approaches can be studied as a future direction. Hence, the overall optimization is given as:

$$\{\mathbf{D}^*, \tilde{\mathbf{P}}^*, \tilde{\mathbf{X}}^*\} = \arg\min_{\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}} \mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}) + \lambda \mathcal{C}_2(\tilde{\mathbf{P}})$$
$$\text{s.t. } \mathbf{P_i P_i^T} = \mathbf{I},\ i = 1, 2 \text{ and } \|\tilde{\mathbf{x}}_j\|_0 \leq T_0, \forall j \tag{5}$$

where, λ is a positive constant.

B. Multiple domains

The above formulation can be extended so that it can handle multiple domains. For M domain problem, we simply construct matrices $\tilde{\mathbf{Y}}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}$ as:

$$\tilde{\mathbf{P}} = [\mathbf{P_1}, \cdots, \mathbf{P_M}]\ , \tilde{\mathbf{Y}} = \begin{pmatrix} \mathbf{Y_1} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{Y_M} \end{pmatrix},$$

and

$$\tilde{\mathbf{X}} = [\mathbf{X_1}, \cdots, \mathbf{X_M}].$$

With these definitions, (5) can be extended to multiple domains as follows

$$\{\mathbf{D}^*, \tilde{\mathbf{P}}^*, \tilde{\mathbf{X}}^*\} = \arg\min_{\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}} \mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}) + \lambda \mathcal{C}_2(\tilde{\mathbf{P}})$$
$$\text{s.t. } \mathbf{P_i P_i^T} = \mathbf{I},\ i = 1, \cdots, M \text{ and } \|\tilde{\mathbf{x}}_j\|_0 \leq T_0, \forall j \tag{6}$$

C. Special case of $\mathbf{P_1} = \mathbf{P_2} = \cdots = \mathbf{P_M}$

In the special of domain adaptation, where same features are extracted for all the domains such that $d_1 = d_2 = \cdots = d_M$, and the domain shift is not large (e.g. matching frontal faces to profile faces), same projection matrix can be used for all the domains.

D. Discriminative Dictionary

The dictionary learned in (5) can reconstruct the two domains well, but it cannot discriminate between the data from different classes. Recent advances in learning discriminative dictionaries [34], [49] suggest that learning class-wise, mutually incoherent dictionaries works better for discrimination. To incorporate this into our framework, we write the dictionary $\mathbf{D}$ as $\mathbf{D} = [\mathbf{D_1}, \cdots, \mathbf{D_C}]$, where C is the total number of classes. We modify the cost function similar to [49], which encourages reconstruction samples of a given class by the dictionary of the corresponding class, and penalizes reconstruction by out-of-class dictionaries. The new cost function, $\mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}})$ is given as:

$$\mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}) = \|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}\|_F^2 + \mu\|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}_\mathbf{in}\|_F^2 +$$
$$\nu\|\mathbf{D}\tilde{\mathbf{X}}_\mathbf{out}\|_F^2, \tag{7}$$

where μ and ν are the weights given to the discriminative terms, and matrices $\tilde{\mathbf{X}}_\mathbf{in}$ and $\tilde{\mathbf{X}}_\mathbf{out}$ are given as:

$$\tilde{\mathbf{X}}_\mathbf{in}[i, j] = \begin{cases} \tilde{\mathbf{X}}[i, j], & \mathbf{D_i}, \tilde{\mathbf{Y}}_\mathbf{j} \in \text{ same class} \\ 0, & \text{otherwise,} \end{cases}$$

$$\tilde{\mathbf{X}}_\mathbf{out}[i, j] = \begin{cases} \tilde{\mathbf{X}}[i, j], & \mathbf{D_i}, \tilde{\mathbf{Y}}_\mathbf{j} \in \text{ different class} \\ 0, & \text{otherwise.} \end{cases}$$

The cost function is defined only for labeled data in both domains. Unlabeled data can be handled using semi-supervised approaches to dictionary learning [32]. However, we do not explore it further in this paper. Also, note that we do not need to modify the forms of projection matrices, since they capture the overall domain shift, and hence are independent of class variations.

E. Optimization

The optimization problem (6) is non-convex in the variables $\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}$. Hence, we optimize the cost using alternate minimization strategy, where first $\tilde{\mathbf{P}}$ is updated, keeping $\mathbf{D}$, $\tilde{\mathbf{X}}$ fixed followed by updating $\mathbf{D}$ and $\tilde{\mathbf{X}}$, keeping $\tilde{\mathbf{P}}$ fixed.

- **Updating $\tilde{\mathbf{P}}$:** For fixed $\mathbf{D}, \tilde{\mathbf{X}}$, the optimization can be written as:

$$\tilde{\mathbf{P}}^* = \arg\min_{\tilde{\mathbf{P}}} \|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}\|_F^2 + \mu\|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}_{\mathbf{in}}\|_F^2 - \lambda \text{trace}((\tilde{\mathbf{P}}\tilde{\mathbf{Y}})(\tilde{\mathbf{P}}\tilde{\mathbf{Y}})^T) \text{ s.t. } \mathbf{P_i}\mathbf{P_i^T} = \mathbf{I},\ i = 1, \cdots, M \tag{8}$$

  Howver, this is not a convex problem because of the orthonormality constraints on $\mathbf{P_i}$. Specifically, it involves optimization on Stiefel manifold, hence, we solve it using the manifold optimization technique described in [45].
- **Updating $\mathbf{D}, \tilde{\mathbf{X}}$:** For fixed $\tilde{\mathbf{P}}$ the optimization problem can be written as:

$$\{\mathbf{D}^*, \tilde{\mathbf{X}}^*\} = \arg\min_{\mathbf{D}, \tilde{\mathbf{X}}} \|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}\|_F^2 + \mu\|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}_{\mathbf{in}}\|_F^2 + \nu\|\mathbf{D}\tilde{\mathbf{X}}_{\mathbf{out}}\|_F^2 \text{ s.t. } \|\tilde{\mathbf{x}}_j\|_0 \leq T_0, \forall j \tag{9}$$

  This is discriminative dictionary learning problem, and we use the framework of [49] to update $\mathbf{D}, \tilde{\mathbf{X}}$. This can be easily generalized to utilize other dictionary learning algorithms as well.

The proposed Shared Discriminative Dictionary Learning (SDDL) algorithm is summarized in Algorithm 1.

Input: Data $\{\mathbf{Y_i}\}_{\mathbf{i}=1}^M$ and corresponding class labels $\{C_i\}_{i=1}^M$ for M domains, sparsity level T_0, dictionary size K and dimension d_f, parameter values μ, ν

Procedure:

1. *Initialize:* Initialize $\tilde{\mathbf{P}}$ such that $\mathbf{P_i}\mathbf{P_i} = \mathbf{I}\ \forall\ i = 1, \cdots, M$. For this, PCA of the data, $\mathbf{Y_i}$ can be used to initialize $\mathbf{P_i}$.
2. *Update step for $\tilde{\mathbf{P}}$:* Update $\tilde{\mathbf{P}}$ as:

$$\tilde{\mathbf{P}}^* = \arg\min_{\tilde{\mathbf{P}}} \|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}\|_F^2 + \mu\|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}_{\mathbf{in}}\|_F^2 - \lambda \text{trace}((\tilde{\mathbf{P}}\tilde{\mathbf{Y}})(\tilde{\mathbf{P}}\tilde{\mathbf{Y}})^T) \text{ s.t. } \mathbf{P_i}\mathbf{P_i^T} = \mathbf{I},\ i = 1, \cdots, M$$

   using Stiefel manifold optimization technique [45].
3. *Update step for $\mathbf{D}, \tilde{\mathbf{X}}$:* Learn common dictionary $\mathbf{D}$ and sparse code, $\tilde{\mathbf{X}}$ using discriminative dictionary learning algorithm such as FDDL [49]

$$\{\mathbf{D}^*, \tilde{\mathbf{X}}^*\} = \arg\min_{\mathbf{D}, \tilde{\mathbf{X}}} \|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}\|_F^2 + \mu\|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}_{\mathbf{in}}\|_F^2 + \nu\|\mathbf{D}\tilde{\mathbf{X}}_{\mathbf{out}}\|_F^2 \text{ s.t. } \|\tilde{\mathbf{x}}_j\|_0 \leq T_0, \forall j$$

Output: Learned dictionary $\mathbf{D}$, projection matrices $\{\mathbf{P_i}\}_{\mathbf{i}=1}^M$

Algorithm 1: Shared Domain-adapted Dictionary Learning (SDDL)

IV. Non-linear extension

In many vision problems, projecting the original features may not be good enough due to non-linearity in data. This can be overcome by transforming the data into a high-dimensional feature space. Let $\mathbf{\Phi} : \mathbb{R}^n \rightarrow \mathcal{H}$ be a mapping to the reproducing kernel Hilbert space $\mathcal{H}$. The mapping $\boldsymbol{\mathcal{P}}_\mathbf{i}$ to the reduced space, can be characterized by a compact, linear operator, $\boldsymbol{\mathcal{P}}_\mathbf{i} : \mathcal{H} \rightarrow \mathbb{R}^d$. As the feature space can be infinite dimensional, the projection matrix $\boldsymbol{\mathcal{P}}_\mathbf{i}$ cannot be handled in this form. To make the kernelization of the algorithm possible, we use the following proposition:

Proposition 1: *There exists an optimal solution $\mathbf{P_1^*}, \cdots, \mathbf{P_M^*}, \mathbf{D}^*$ to equation (6), which has the following form:*

$$\mathbf{P_i^*} = (\mathbf{Y_i}\mathbf{A_i})^\mathbf{T}\ \forall\ i = 1, \cdots, M \tag{10}$$

$$\mathbf{D}^* = \tilde{\mathbf{P}}^*\tilde{\mathbf{Y}}\tilde{\mathbf{B}} \tag{11}$$

where, $\tilde{\mathbf{P}}^* = [\mathbf{P_1^*}, \cdots, \mathbf{P_M^*}]$, for some $\mathbf{A_i} \in \mathbb{R}^{N_i \times n}$ and some $\tilde{\mathbf{B}} \in \mathbb{R}^{\sum N_i \times K}$.

Proof: See Appendix I.

With this proposition, the cost functions can be written as:

$$\mathcal{C}_1(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{X}}) = \|\tilde{\mathbf{A}}^\mathbf{T}\tilde{\mathbf{K}}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})\|_F^2 + \mu\|\tilde{\mathbf{A}}^\mathbf{T}\tilde{\mathbf{K}}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})\|_F^2 + \nu\|\tilde{\mathbf{A}}^\mathbf{T}\tilde{\mathbf{K}}\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}}\|_F^2 \tag{12}$$

$$\mathcal{C}_2(\tilde{\mathbf{A}}) = -\text{trace}((\tilde{\mathbf{A}}^\mathbf{T}\tilde{\mathbf{K}})(\tilde{\mathbf{A}}^\mathbf{T}\tilde{\mathbf{K}})^T) \tag{13}$$

where, $\tilde{\mathbf{K}} = \tilde{\mathbf{Y}}^\mathbf{T}\tilde{\mathbf{Y}}$ and $\tilde{\mathbf{A}}^\mathbf{T} = [\mathbf{A_1^T}, \cdots, \mathbf{A_M^T}]$. The equality constraints now become:

$$\mathbf{P_i}\mathbf{P_i^T} = \mathbf{A_i^T}\mathbf{K_i}\mathbf{A_i} = \mathbf{I},\ \forall i\ =\ 1, \cdots, M \tag{14}$$

where, $\mathbf{K_i} = \mathbf{Y_i^T}\mathbf{Y_i}$. The optimization problem now becomes:

$$\{\tilde{\mathbf{A}}^*, \tilde{\mathbf{B}}^*, \tilde{\mathbf{X}}^*\} = \arg\min_{\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{X}}} \mathcal{C}_1(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{X}}) + \lambda\mathcal{C}_2(\tilde{\mathbf{A}})$$

$$\text{s.t. } \mathbf{A_i^T}\mathbf{K_i}\mathbf{A_i} = \mathbf{I},\ i = 1, \cdots, M \text{ and } \|\tilde{\mathbf{x}}_j\|_1 \leq T_0, \forall j \tag{15}$$

This formulation allows joint update of $\mathbf{D}$ and $\mathbf{P_i}$ via $\mathbf{A_i}$.

Let $\mathcal{K} = \langle \Phi(\tilde{\mathbf{Y}}), \Phi(\tilde{\mathbf{Y}}) \rangle_\mathcal{H}$. Then, it can be shown similar to proposition 1 that:

$$\boldsymbol{\mathcal{P}}_\mathbf{i}^* = \mathbf{A}^\mathbf{T}\mathbf{\Phi}(\mathbf{Y})^\mathbf{T}; \mathbf{D}^* = \tilde{\mathbf{A}}^\mathbf{T}\mathcal{K}\tilde{\mathbf{B}}.$$

Thus, we get the cost functions as:

$$\mathcal{C}_1(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{X}}) = \|\tilde{\mathbf{A}}^\mathbf{T}\mathcal{K}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}\|_F^2 + \mu\|\tilde{\mathbf{A}}^\mathbf{T}\mathcal{K}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}}\|_F^2 + \nu\|\tilde{\mathbf{A}}^\mathbf{T}\mathcal{K}\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}}\|_F^2, \tag{16}$$

$$\mathcal{C}_2(\tilde{\mathbf{A}}) = -\text{trace}((\tilde{\mathbf{A}}^\mathbf{T}\mathcal{K})(\tilde{\mathbf{A}}^\mathbf{T}\mathcal{K})^T) \tag{17}$$

and the equality constraints as,

$$\mathbf{A_i^T}\mathcal{K}_\mathbf{i}\mathbf{A_i} = \mathbf{I}\quad \forall\ i = 1, \cdots, M,$$

where $\mathcal{K}_i = \langle \Phi(\mathbf{Y_i}), \Phi(\mathbf{Y_i}) \rangle_\mathcal{H}$.

A. Update step for $\tilde{\mathbf{A}}$

Here we assume that $(\tilde{\mathbf{B}}, \tilde{\mathbf{X}})$ are fixed. Then, the optimization for $\tilde{\mathbf{A}}$ can be solved efficiently. We have the following proposition.

Proposition 2: *The optimal solution of equation (40) when $(\tilde{\mathbf{B}}, \tilde{\mathbf{X}})$ are fixed is:*

$$\tilde{\mathbf{A}}^* = \mathbf{V}\mathbf{S}^{-\frac{1}{2}}\mathbf{G}^* \tag{18}$$

where, $\mathbf{V}$ *and* $\mathbf{S}$ *come from the eigendecomposition of* $\tilde{\mathbf{K}} = \mathbf{V}\mathbf{S}\mathbf{V}^{\mathbf{T}}$*, and* $\mathbf{G}^* \in \mathbb{R}^{\sum N_i \times n} = [\mathbf{G_1^{*T}}, \cdots, \mathbf{G_M^{*T}}]^T$ *is the optimal solution of the following problem:*

$$\{\mathbf{G}^*\} = \arg\min_{\mathbf{G}} \text{trace}[\mathbf{G}^{\mathbf{T}}\mathbf{H}\mathbf{G}]$$
$$s.t.\ \mathbf{G_i^T G_i} = \mathbf{I}\ \forall\ i\ = 1, \cdots, M \tag{19}$$

where,

$$\mathbf{H} = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^{\mathbf{T}}((\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})^{\mathbf{T}} + \mu(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})$$
$$(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})^{\mathbf{T}} + \nu(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})^{\mathbf{T}} - \lambda\mathbf{I})\mathbf{V}\mathbf{S}^{\frac{1}{2}} \tag{20}$$

Proof: See Appendix I.

Equation (41) is non-convex due to non-linear equality constraints. Specifically, due to the orthonormality condition on $\mathbf{G_i}$, it involves optimization on the Stiefel manifold. We solved this problem using the efficient approach presented in [45].

B. Update step for $\tilde{\mathbf{B}}, \tilde{\mathbf{X}}$

For a fixed $\tilde{\mathbf{A}}$, the problem becomes that of discriminative dictionary learning, with data as $\mathbf{Z} = \tilde{\mathbf{A}}^{\mathbf{T}}\tilde{\mathbf{K}}$ and dictionary $\mathbf{D} = \tilde{\mathbf{A}}^{\mathbf{T}}\tilde{\mathbf{K}}\tilde{\mathbf{B}}$. To jointly learn the dictionary, $\mathbf{D}$, and sparse code, $\tilde{\mathbf{X}}$, we use the framework of the discriminative dictionary learning approach presented in [49]. Once the dictionary, $\mathbf{D}$, is learned, we can update $\tilde{\mathbf{B}}$ as:

$$\tilde{\mathbf{B}} = \mathbf{Z}^{\dagger}\mathbf{D}, \tag{21}$$

where $\mathbf{Z}^{\dagger}$ is the pseudo-inverse of $\mathbf{Z}$ defined as $\mathbf{Z}^{\dagger} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T$.

The proposed, Non-linear Shared Domain-adapted Dictionary Learning (kerSDDL) algorithm is summarized in Algorithm 2.

Input: Data $\{\mathbf{Y_i}\}_{\mathbf{i}=1}^{M}$ and corresponding class labels $\{C_i\}_{i=1}^{M}$ for M domains, sparsity level T_0, dictionary size K and dimension n, parameter values μ, ν

Procedure:

1. *Initialize:* Initialize $\tilde{\mathbf{A}}$ such that $\mathbf{A_i}\mathcal{K}_\mathbf{i}\mathbf{A_i} = \mathbf{I}$ $\forall\ i = 1, \cdots, M$. For this, find SVD of each kernel matrix, $\mathcal{K}_\mathbf{i} = \mathbf{V_i}\mathbf{S_i}\mathbf{V_i^T}$. Set $\mathbf{A}_i$ as the matrix of eigen-vectors with top n eigen-values as columns.
2. *Update step for* $\tilde{\mathbf{B}}$*:* Learn common dictionary $\mathbf{D}$ with data as $\mathbf{Z} = \tilde{\mathbf{A}}^{\mathbf{T}}\mathcal{K}$, and using discriminative dictionary learning algorithm as FDDL. Update $\tilde{\mathbf{B}}$ as:

$$\tilde{\mathbf{B}} = \mathbf{Z}^{\dagger}\mathbf{D}$$

3. *Update step for* $\tilde{\mathbf{A}}$*:* Update $\tilde{\mathbf{A}}$ as:

$$\{\mathbf{G}^*\} = \arg\min_{\mathbf{G}} \text{trace}[\mathbf{G}^{\mathbf{T}}\mathbf{H}\mathbf{G}]$$
$$s.t.\ \mathbf{G_i^T G_i} = \mathbf{I}\ \forall\ i\ = 1, \cdots, M$$

where, $\tilde{\mathbf{A}}^* = \mathbf{V}\mathbf{S}^{-\frac{1}{2}}\mathbf{G}^*$ and $\mathbf{H}$ is:

$$\mathbf{H} = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^{\mathbf{T}}((\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})^{\mathbf{T}} + \mu(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})$$
$$(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})^{\mathbf{T}} + \nu(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})^{\mathbf{T}} - \lambda\mathbf{I})\mathbf{V}\mathbf{S}^{\frac{1}{2}}$$

Output: Learned dictionary $\mathbf{D}$, projection matrices $\{\mathbf{A_i}\}_{\mathbf{i}=1}^{M}$

Algorithm 2: Non-linear Shared Domain-adapted Dictionary Learning (kerSDDL)

V. Classification

Given a test sample, $\mathbf{y_{te}}$ from domain k, we propose the following steps for classification, similar to [28].

A. Linear Classification

1) Compute the embedding of the sample in the common subspace, $\mathbf{z_{te}}$ using the projection, $\boldsymbol{P}_\mathbf{k}^*$.

$$\mathbf{z_{te}} = \mathbf{P_k}^*\mathbf{y_{te}}$$

2) Compute the sparse coefficients, $\hat{\mathbf{x}}_\mathbf{te}$, of the embedded sample over dictionary $\mathbf{D}$ using the OMP algorithm [31].

$$\hat{\mathbf{x}}_\mathbf{te} = \arg\min_{\mathbf{x}} \|\mathbf{z_{te}} - \mathbf{D}\mathbf{x}\|_\mathbf{F}^\mathbf{2}\ \text{s.t.}\ \|\mathbf{x}\|_0 \leq T_0.$$

3) Now, the sample can be assigned to class i, if the reconstruction using the class dictionary, $\mathbf{D_i}$ and the sparse code corresponding to the atoms of the dictionary, $\hat{\mathbf{x}}_\mathbf{te}^\mathbf{i}$ is minimum.

$$\text{Output class} = \arg\min_{i=1,\cdots,C} \|\mathbf{z_{te}} - \mathbf{D_i}\hat{\mathbf{x}}_\mathbf{te}^\mathbf{i}\|_\mathbf{F}^\mathbf{2}.$$

However, the reconstruction error may not be discriminative enough in the reduced space. So, we project the class-wise reconstruction, $\mathbf{D_i}\hat{\mathbf{x}}_\mathbf{te}^\mathbf{i}$ into the feature space, and assign the test sample to the class with the minimum error in the original feature space:

$$\text{Output class} = \arg\min_{i=1,\cdots,C} \|\mathbf{y_{te}} - \mathbf{P_k}^{*\mathbf{T}}\mathbf{D_i}\hat{\mathbf{x}}_\mathbf{te}^\mathbf{i}\|_\mathbf{F}^\mathbf{2} \tag{22}$$

B. Non-linear classification

Here, we consider the general case of classifying mapping of the sample into kernel space, $\boldsymbol{\Phi}(\mathbf{y_{te}})$.

1) Compute the embedding of the sample in the common subspace, $\mathbf{z_{te}}$ using the projection, $\boldsymbol{\mathcal{P}}_\mathbf{k}^*$.

$$\mathbf{z_{te}} = \boldsymbol{\mathcal{P}}_\mathbf{k}^*\boldsymbol{\Phi}(\mathbf{y_{te}}) = \mathbf{A_k}\mathcal{K}_\mathbf{te}$$

where, $\mathcal{K}_\mathbf{te} = \langle\boldsymbol{\Phi}(\mathbf{Y_k}), \boldsymbol{\Phi}(\mathbf{y_{te}})\rangle$.

2) Compute the sparse coefficients, $\hat{\mathbf{x}}_\mathbf{te}$, of the embedded sample over dictionary $\mathbf{D}$ using the OMP algorithm [31].

$$\hat{\mathbf{x}}_\mathbf{te} = \arg\min_{\mathbf{x}} \|\mathbf{z_{te}} - \mathbf{D}\mathbf{x}\|_\mathbf{F}^\mathbf{2}\ \text{s.t.}\ \|\mathbf{x}\|_0 \leq T_0.$$

3) Project the class-wise reconstruction, $\mathbf{D_i}\hat{\mathbf{x}}_\mathbf{te}^\mathbf{i}$ into the feature space, and assign the test sample to the class with the minimum error in the original feature space:

$$\text{Output class} = \arg\min_{i=1,\cdots,C} \|\boldsymbol{\Phi}(\mathbf{y_{te}}) - \boldsymbol{\mathcal{P}}_\mathbf{k}^{*\mathbf{T}}\mathbf{D_i}\hat{\mathbf{x}}_\mathbf{te}^\mathbf{i}\|_\mathbf{F}^\mathbf{2}$$
$$= \arg\min_{i=1,\cdots,C} \boldsymbol{\kappa}_\mathbf{te} - \mathbf{2}\mathcal{K}_\mathbf{te}\mathbf{A_k^*D_i} + \hat{\mathbf{x}}_\mathbf{te}^\mathbf{iT}\mathbf{D_i^T}\mathbf{A_k^*}\mathcal{K}_\mathbf{k}\mathbf{A_k^*D_i}\hat{\mathbf{x}}_\mathbf{te}^\mathbf{i},$$

where $\boldsymbol{\kappa}_\mathbf{te} = \langle\boldsymbol{\Phi}(\mathbf{y_{te}}), \boldsymbol{\Phi}(\mathbf{y_{te}})\rangle$.

VI. Experiments

We conducted various experiments to ascertain the effectiveness of the proposed method. First, we demonstrate

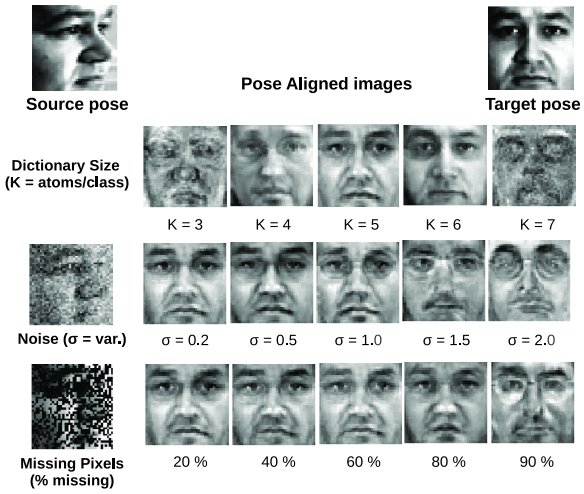
some synthesis and recognition results on the CMU Multi-Pie dataset for face recognition across pose and illumination variations. This also provides insights into our method through visual examples. Next we show the performance of our method on domain adaptation databases and compare it with existing adaptation algorithms.

A. CMU Multi-Pie Dataset

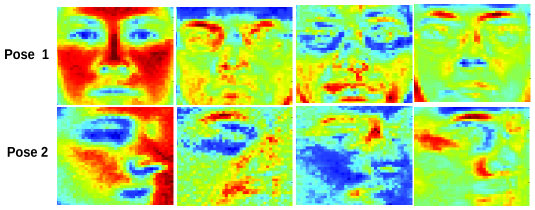
The Multi-pie dataset [19] is a comprehensive face dataset of 337 subjects, having images taken across 15 poses, 20 illuminations, 6 expressions and 4 different sessions. For the purpose of our experiment, we used 129 subjects common to both Session 1 and 2. The experiment was done on 5 poses, ranging from frontal to 75º. Frontal faces were taken as the source domain, while different off-frontal poses were taken as target domains. Dictionaries were trained using illuminations $\{1, 4, 7, 12, 17\}$ from the source and the target poses, in Session 1 per subject. All the illumination images from Session 2, for the target pose, were taken as probe images. The linear kernel was used for all the experiments.

1) Pose Alignment: First we consider the problem of pose alignment using the proposed dictionary learning framework. Pose alignment is challenging due to the highly non-linear changes induced by 3-D rotation of face. Images at the extreme pose of 60º were taken as the target pose. A shared discriminative dictionary was learned using the approach described in this paper. Given the probe image, it was projected on the latent subspace and reconstructed using the dictionary. The reconstruction was back-projected onto the source pose domain, to give the aligned image. Figure 2(a) shows the synthesized images for various conditions. We can draw some useful insights about the method from this figure. Firstly, it can be seen that there is an optimal dictionary size, $K = 5$, where the best alignment is achieved. Further, by learning a discriminative dictionary, the identity of the subject is retained. For $K = 7$, the alignment is not good, as the learned dictionary is not able to successfully correlate the two domains when there are more atoms in the dictionary. Dictionary with $K = 3$ has higher reconstruction error, hence the result is not optimal. We chose $K = 5$ for additional experiments with noisy images. It can be seen that from rows 2 and 3 that the proposed method is robust even at high levels of noise and missing pixels. Moreover, de-noised and in-painted synthesized images are produced as shown in rows 2 and 3 of Figure 2(a), respectively. This shows the effectiveness of our method. Moreover, the learned projection matrices (Figure 2(b)) show that our method can learn the internal structure of the two domains. As a result, it is able to learn a robust common dictionary.

2) Recognition: We also conducted recognition experiment using the set-up described above. Table I shows that our method compares favorably with some of the recently proposed multi-view recognition algorithms [38], and gives the best performance on average. The linear kernel was found to be giving better performance, hence, we do not report the results for kerSDDL. The dictionary learning algorithm, FDDL [49] is not optimal here as it is not able to efficiently represent the non-linear changes introduced by the pose variation.



(a)



(b)

Fig. 2. (a) Examples of pose-aligned images using the proposed method. Synthesis in various conditions demonstrate the robustness of the method. (b) First few components of the learned projection matrices for the two poses.

Method	Probe pose					Average
	15º	30º	45º	60º	75º	
PCA	15.3	5.3	6.5	3.6	2.6	6.7
PLS [37]	39.3	40.5	41.6	41.1	38.7	40.2
LDA	98.0	94.2	91.7	84.9	79.0	89.5
CCA [37]	92.1	89.7	88.0	86.1	83.0	83.5
GMLDA [38]	**99.7**	**99.2**	98.6	94.9	95.4	97.6
FDDL [49]	96.8	90.6	94.4	91.4	90.5	92.7
SDDL	98.4	98.2	**98.9**	**99.1**	**98.8**	**98.7**

TABLE I
COMPARISON OF THE PROPOSED METHOD WITH OTHER ALGORITHMS FOR FACE RECOGNITION ACROSS POSE.

B. Object Recognition

We now evaluate our method for object recognition. The experiments use the dataset which was introduced in [35]. The dataset consists of images from 3 sources: Amazon (consumer images from online merchant sites), DSLR (images by DSLR camera) and Webcam (low quality images from webcams). In addition, we also tested on the Caltech-256 dataset [18], taking it as the fourth domain. Figure 3 shows sample images from these datasets, and clearly highlights the differences between the domains. We follow 2 set-ups for testing the algorithm. In the first set-up, 10 common classes: BACKPACK, TOURING-BIKE, CALCULATOR, HEADPHONES, COMPUTER- KEYBOARD, LAPTOP-101, COMPUTER- MONITOR, COMPUTER-MOUSE, COFFEE- MUG, AND VIDEO- PROJECTOR, common to all the four datasets are used. In this

Fig. 3. Example images from KEYBOARD and BACK-PACK categories in Caltech-256, Amazon, Webcam and DSLR. Caltech-256 and Amazon datasets have diverse images, Webcam and DSLR are similar datasets with mostly images from offices.

case, there are a total of 2533 images. Each category has 8 to 151 images in a dataset. In the second set-up, we evaluate the methods for adaptation using multiple domains. In this case, we restrict to the first dataset, and test on all the 31 classes in it. For both the cases, we use 20 training samples per class for Amazon/Caltech, and 8 samples per class for DSLR/Webcam when used as source, and 3 training samples for all of them when used for target domain. Rest of the data in the target domain is used for testing. The experiment is run 20 times for random train/test splits and the result is averaged over all the runs.

We demonstrate the effectiveness of the proposed method for two cases: 1. same features extracted for all the domains, 2. different features extracted for different domains.

1) Adaptation with same features: First, we test the proposed algorithms for the case when the same feature is extracted for all the domains.

Feature Extraction: We used the 800-bin SURF features provided by [35] for the Amazon, DSLR and Webcam datasets. For the Caltech images, first SURF features were extracted from the images of the Caltech data and a random subset of the Amazon dataset. The features obtained from the Amazon dataset were grouped into 800 clusters using the k-means algorithm. The cluster centers were then used to quantize the SURF features obtained from the Caltech data to form 800-bin histograms. The histograms were normalized and then used for classification.

Parameter Settings: We set $\mu = 4$ and $\nu = 30$. Dictionary size, $K = 4$ atoms per class and final dimension, $n = 60$ for the first set-up, for both SDDL and kerSDDL algorithms. For the second set-up, $K = 6$ atoms per class and $n = 90$ for SDDL and kerSDDL. For FDDL, the parameters, μ and ν are the same as SDDL, and we learn $K = 8$ atoms per class for the first set-up and $K = 10$ atoms per class for the second. The SDDL algorithm was trained using same projection matrix for all the domains as discussed in Section III-C. We initialized the matrices as PCA of source, target data or both data taken together, and report the best performance among them. For kerSDDL method, we used the simple non-parametric histogram intersection kernel for reporting all the values. The projection matrix for kerSDDL was initialized as described in Algorithm 2. The FDDL dictionary was trained using both the source and the target domain features, as it was found to give the best results. Original histogram features were used for both the algorithms. Performance of the proposed SDDL method is compared to FDDL [49], and some recently proposed domain-adaptation algorithms [35], [16], [17], [15], [21], [25], [29].

1) **Results using single source:** Table II(a) shows a comparison of the results of different methods on 8 source-target pairs. The proposed algorithms give the best performance for 6 domain pairs, and is the second best for 2 pairs. For Caltech-DSLR and Amazon-Webcam domain pairs, there is more than 15% improvement over the GFK [15] and SID [29] algorithms. Furthermore, a comparison with the FDDL algorithm shows that the learning framework of [49] is inefficient, when the test data comes from a different distribution than the data used for training. Both the SDDL and kerSDDL algorithms perform better than FDDL on all the pairs.
2) **Results using multiple sources:** As our proposed framework can also handle multiple domains, we also experimented with multiple source adaptation. Table II (b) shows the results for 3 possible combinations. The proposed methods outperforms the original SGF method [16] on two settings, and other methods for all the settings. However, [17] reports higher numbers on webcam and amazon as targets, using boosted classifiers. Similarly techniques can be explored for improving the proposed method as a future direction.
3) **Ease of adaptation:** A rank of domain (ROD) metric was introduced in [15] to measure the adaptability of different domains. It was shown that ROD correlates with the performance of adaptation algorithm. For example, Amazon-Webcam pair has higher ROD than DSLR-Webcam pair, hence, GFK performs worse on the former. However, for our case, we find that the recognition rates for these cases are 72.0 % and 72.6 %, respectively. This is the case because by learning projections along-with the common dictionary, we can achieve a better alignment of the datasets.
4) **Parameter Variations:** We also conducted experiments studying recognition performance under different input parameters. Figure 4 shows the result of different settings. The implications are briefly discussed below:
    a) **Number of source images:** Here, we choose Amazon/Webcam domain pair, as it is "difficult" to adapt. We increased the number of source images and studied the performance of SDDL and kerSDDL and compared it with FDDL. It can be seen that while FDDL's performance decreases sharply with more source images, SDDL and kerSDDL

(a) Performance comparison on single source four domains benchmark (C: caltech, A: amazon, D: dslr, W: webcam)

Methods	C → A	C → D	A → C	A → W	W → C	W → A	D → A	D → W
Metric[35]	33.7 ± 0.8	35.0 ± 1.1	27.3 ± 0.7	36.0 ± 1.0	21.7 ± 0.5	32.3 ± 0.8	30.3 ± 0.8	55.6 ± 0.7
SGF[16]	40.2 ± 0.7	36.6 ± 0.8	37.7 ± 0.5	37.9 ± 0.7	29.2 ± 0.7	38.2 ± 0.6	39.2 ± 0.7	69.5 ± 0.9
GFK[15]	46.1 ± 0.6	55.0 ± 0.9	39.6 ± 0.4	56.9 ± 1.0	32.8 ± 0.1	46.2 ± 0.6	46.2 ± 0.6	80.2 ± 0.4
HFA [25]	45.5 ± 0.9	51.9 ± 1.1	31.1 ± 0.6	58.6 ± 1.0	31.1 ± 0.6	45.9 ± 0.7	45.8 ± 0.9	62.1 ± 0.7
SID [29]	50 ± 0.5	57.1 ± 0.4	41.5 ± 0.8	57.8 ± 0.5	40.6 ± 0.4	$\mathbf{51.5 \pm 0.6}$	50.3 ± 0.2	$\mathbf{87.8 \pm 1.0}$
FDDL[49]	39.3 ± 2.9	55.0 ± 2.8	24.3 ± 2.2	50.4 ± 3.5	22.9 ± 2.6	41.1 ± 2.6	36.7 ± 2.5	65.9 ± 4.9
SDDL	$\mathbf{54.4 \pm 2.2}$	67.7 ± 4.0	$\mathbf{41.8 \pm 2.2}$	67.1 ± 3.2	$\mathbf{41.5 \pm 2.1}$	48.2 ± 2.3	$\mathbf{50.6 \pm 2.1}$	86.4 ± 2.8
kerSDDL	49.5 ± 2.6	$\mathbf{76.7 \pm 3.9}$	27.4 ± 2.4	$\mathbf{72.0 \pm 4.8}$	29.7 ± 1.9	49.4 ± 2.1	48.9 ± 3.8	72.6 ± 2.1

(b) Performance comparison on multiple sources three domains benchmark

Source	Target	SGF* [17]	SGF [16]	RDALR[21]	FDDL[49]	SDDL	kerSDDL
dslr, amazon	webcam	$\mathbf{64.5 \pm 0.3}$	52 ± 2.5	36.9 ± 1.1	41.0 ± 2.4	53.6 ± 1.2	57.8 ± 2.4
amazon, webcam	dslr	51.3 ± 0.7	39 ± 1.1	31.2 ± 1.3	38.4 ± 3.4	55.8 ± 2.0	$\mathbf{56.7 \pm 2.3}$
webcam, dslr	amazon	$\mathbf{38.4 \pm 1.0}$	28 ± 0.8	20.9 ± 0.9	19.0 ± 1.2	23.8 ± 1.2	24.1 ± 1.6

TABLE II
COMPARISON OF THE PERFORMANCE OF THE PROPOSED METHOD ON THE AMAZON, WEBCAM, DSLR AND CALTECH DATASETS.

methods show increase in the performance. Hence, by adapting the source to the target domain, our method can use the source information to increase the accuracy of target recognition, even when their distributions are very different.

b) **Dictionary size:** We varied the dictionary size for kerSDDL algorithm for different source-target pairs. All the domain pairs show an initial sharp increase in the performance, and then become almost flat after the dictionary size of 3 or 4. The flat region indicates that alignment of the source and the target data is limited by the number of available target samples. But also, on a positive note, it can be seen that even a smaller dictionary can give the optimal performance.

c) **Common subspace dimension:** Similar to the previous case, we get an initial sharp increase followed by a flat recognition curve. This shows that the method is effective even when the data is projected onto a low-dimensional space.

5) **Convergence:** Figure 4(d) shows the cost function with iteration for SDDL and kerSDDL algorithms. It can be seen that both the algorithms converge quickly in 5-6 iterations.

2) Adaptation with different features: The proposed methods can be generalized for cases when features of different types (like dimension) are extracted for different domains. Note that the original FDDL algorithm [49] cannot be used for such cases. Also some of the adaptation algorithms compared above cannot be generalized for such cases [15], [17], [29], [21]. We compare the proposed methods with recent heterogeneous adaptation methods [25], [24], [43], [40] and demonstrate their effectiveness.

Experiment Set-up: We restrict the evaluation to Amazon, DSLR and Webcam datasets, using all the 31 classes for evaluation. The train-test split was done as described in Section VI-B1. The evaluation was done using 3 different experimental set-ups described as follows:

1) **DSLR-600 dataset:** We extracted 600-dimensional SURF features for DSLR dataset as described in [24]. We demonstrate results for adaptation from 800-dimensional SURF features extracted in Section VI-B1 to the new features.

2) **Halftone and Sketch datasets:** To test the proposed algorithms across different domain shifts, we created two new datasets by half-toning and edge detection from the original dataset. Figure 5 shows some of the images from these datasets. Half-toning images, which imitate the effect of jet-printing technology in the past, were generated using the dithering algorithm in [27]. Edge images are obtained by applying the Canny edge detector [4] with the threshold set to 0.07. We extracted 800-bin SURF features for both the datasets, following the same approach as for the original dataset.

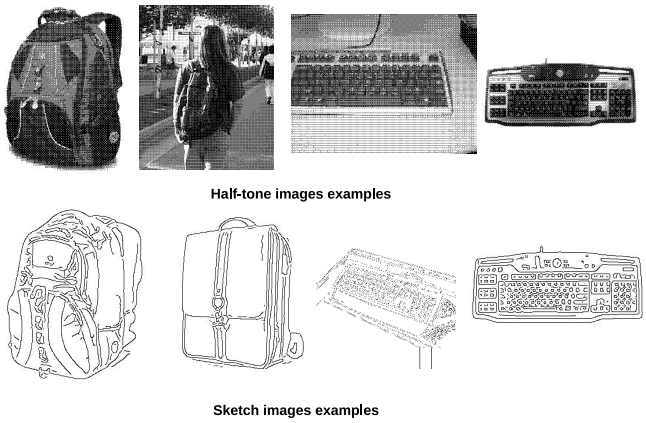


Fig. 5. Example images from half-tone and sketch datasets.

Parameter Setting: We set $\mu = 4$ and $\nu = 30$. Dictionary size, $K = 4$ atoms per class and final dimension, $n = 90$ for all the set-ups, for both SDDL and kerSDDL algorithms. For kerSDDL method, we used the non-parametric histogram intersection kernel for all the experiments. The projection matrix for kerSDDL was initialized as described in Algorithm 2. For SDDL, we initialized separate projection matrix for each domain as described in Algorithm 1.

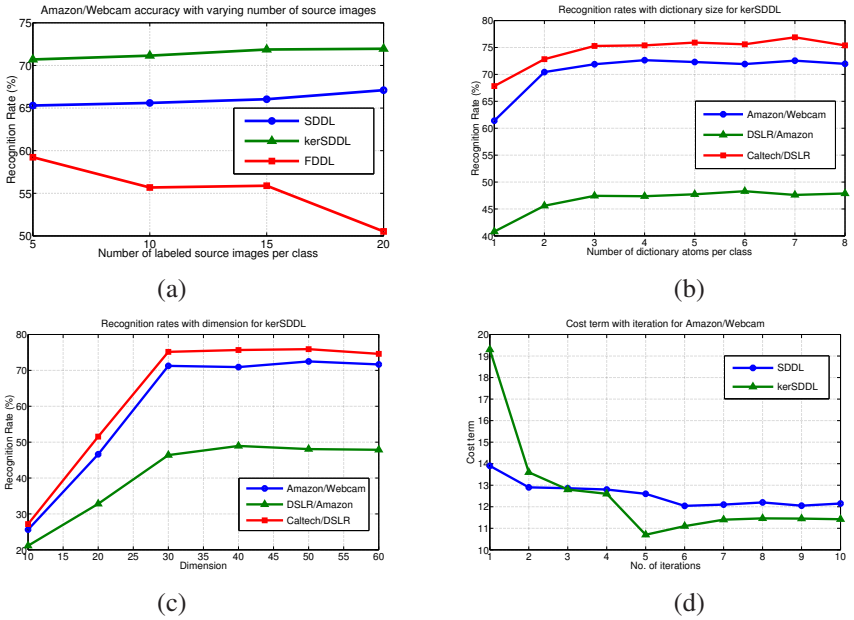


Fig. 4. Recognition performance under different: (a) number of source images, (b) dictionary size, (c) common subspace dimension. (d) Convergence of the proposed algorithms. Naming of domains is done as source/target.

1) **DSLR-600 adaptation** Table II(a) shows the comparison of the proposed methods for adaptation of 800-dimensinal SURF features to 600-dimensional SURF features from DSLR data. It can be seen that kerSDDL gives better performance than the recent state-of-art heterogenous adaptation methods. SDDL algorithm also performs on par with other algorithms.
2) **Half-tone and Sketch dataset adaptation** Tables II(b), II(c) show results for adaptation from original images to half-tone and sketch image datasets respectively. The proposed algorithms are compared with [24] and nearest neighbor classification method. It can be seen that the kerSDDL algorithm performs better than [24] for all the source-target pairs.

VII. Conclusion

We have proposed a novel framework for adapting dictionaries to testing domains under arbitrary domain shifts. An efficient optimization method is presented. Furthermore, the method is kernelized so that it is robust and can deal with the non-linearity present in the data. The learned dictionary is compact and low-dimensional. To gain intuition into the working of the method, we demonstrate applications like pose alignment and pose-robust face recognition. We evaluate the proposed algorithms for different object recognition adaptations. Specifically, we show that the methods can be used for cases like heteregenous domain adapation, where original dictionary learning framework cannot be applied. The proposed methods were compared with the recent domain adaptation algorithms, and the proposed methods were found to be better or comparable to the previous methods. Future works will include studying the effect of using unlabeled data while training, and other relevant problems like large-scale and online adaptation of dictionaries.

VIII. Acknowledgment

This work was partially supported by an ONR grant N00014-12-1-0124.

IX. Appendix

The optimization problem (6) is given as:

$$\{\mathbf{D}^*, \tilde{\mathbf{P}}^*, \tilde{\mathbf{X}}^*\} = \arg\min_{\mathbf{D},\tilde{\mathbf{P}},\tilde{\mathbf{X}}} \mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}) + \lambda \mathcal{C}_2(\tilde{\mathbf{P}})$$
$$\text{s.t. } \mathbf{P_i}\mathbf{P_i^T} = \mathbf{I},\ i = 1, \cdots, M \text{ and } \|\tilde{\mathbf{x}}_j\|_1 \leq T_0, \forall j \quad (23)$$

where,

$$\mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}}) = \|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}\|_F^2 + \mu\|\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}_{\mathbf{in}}\|_F^2 + \nu\|\mathbf{D}\tilde{\mathbf{X}}_{\mathbf{out}}\|_F^2, \quad (24)$$
$$\mathcal{C}_2(\tilde{\mathbf{P}}) = -\text{trace}((\tilde{\mathbf{P}}\tilde{\mathbf{Y}})(\tilde{\mathbf{P}}\tilde{\mathbf{Y}})^T). \quad (25)$$

A. Proposition 1:

There exists an optimal solution $\mathbf{P_1^*}, \cdots, \mathbf{P_M^*}, \mathbf{D}^*$ *to equation (6), which has the following form:*

$$\mathbf{P_i^*} = (\mathbf{Y_i}\mathbf{A_i})^{\mathbf{T}}\ \forall\ i = 1, \cdots, M \quad (26)$$
$$\mathbf{D}^* = \tilde{\mathbf{P}}^*\tilde{\mathbf{Y}}\tilde{\mathbf{B}} \quad (27)$$

Proof:

Form for $\mathbf{D}^*$: First we will show the form for $\mathbf{D}^*$. We can decompose $\mathbf{D}^*$ into two orthogonal components as follows:

$$\mathbf{D}^* = \mathbf{D}_{\|} + \mathbf{D}_{\perp} \quad (28)$$
$$\text{where, } \mathbf{D}_{\|} = (\tilde{\mathbf{P}}\tilde{\mathbf{Y}})\tilde{\mathbf{B}},\ \mathbf{D}_{\perp}^{\mathbf{T}}(\tilde{\mathbf{P}}\tilde{\mathbf{Y}}) = \mathbf{0} \quad (29)$$

(a) Performance comparison on recognition across different features

Source	Target	Metric-asymm [24]	HeMap [40]	DAMA [43]	HFA [25]	SDDL	kerSDDL
amazon	dslr-600	53.1 ± 2.4	42.8 ± 2.4	53.3 ± 2.4	55.4 ± 2.8	50.4 ± 2.5	$\mathbf{61.5 \pm 3.6}$
webcam	dslr-600	53.0 ± 3.2	42.2 ± 2.6	53.2 ± 3.2	54.3 ± 3.7	49.4 ± 2.9	$\mathbf{58.3 \pm 2.6}$

(b) Performance comparison for adaptation to half-tone images

Methods	W $\rightarrow$ D-half	D $\rightarrow$ W-half	A $\rightarrow$ D-half	A $\rightarrow$ W-half
kNN	25.2 ± 2.6	35.2 ± 2.2	25.0 ± 2.0	34.0 ± 1.4
Metric[35]	38.8 ± 2.4	40.2 ± 2.0	33.8 ± 3.8	39.0 ± 2.2
SDDL	32.3 ± 1.7	36.4 ± 1.9	30.1 ± 2.0	34.7 ± 1.7
kerSDDL	$\mathbf{42.0 \pm 2.6}$	$\mathbf{43.0 \pm 2.3}$	$\mathbf{46.4 \pm 3.1}$	$\mathbf{51.0 \pm 2}$

(c) Performance comparison for adaptation to sketch images

Methods	W $\rightarrow$ D-sketch	D $\rightarrow$ W-sketch	A $\rightarrow$ D-sketch	A $\rightarrow$ W-sketch
kNN	31.4 ± 2.7	31.3 ± 1.7	32.1 ± 2.4	33.6 ± 2.7
Metric[35]	39.1 ± 2.7	35.0 ± 2.2	38.0 ± 2.8	37.3 ± 2.5
SDDL	35.8 ± 2.1	32.1 ± 1.8	33.8 ± 2.1	34.0 ± 1.8
kerSDDL	$\mathbf{41.5 \pm 2.6}$	$\mathbf{38.0 \pm 2.6}$	$\mathbf{42.1 \pm 2.4}$	$\mathbf{42.5 \pm 2.3}$

TABLE III
COMPARISON OF THE PERFORMANCE OF THE PROPOSED METHODS FOR PERFORMANCE ON ADAPTATION FOR DSLR-600, HALF-TONE AND SKETCH DATASETS.

for some $\mathbf{B} \in \mathbb{R}^{\sum_{i=1}^{M} N_i \times K}$. Substituting the value of $\mathbf{D}^*$ into the value of $\mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{P}}, \tilde{\mathbf{X}})$, we get for the three terms of $\mathcal{C}_1$, ignoring the multiplicative constants μ, ν:

$$\begin{aligned}&\textbf{First Term} = \text{trace}((\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}})^T(\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}))\\ &= \text{trace}(\tilde{\mathbf{Y}}^\mathbf{T}\tilde{\mathbf{P}}^\mathbf{T}\tilde{\mathbf{P}}\tilde{\mathbf{Y}} + \tilde{\mathbf{Y}}^\mathbf{T}\tilde{\mathbf{P}}^\mathbf{T}\mathbf{D}_\parallel\tilde{\mathbf{X}} + \tilde{\mathbf{X}}^\mathbf{T}\mathbf{D}_\parallel^\mathbf{T}\mathbf{D}_\parallel\tilde{\mathbf{X}} + \\ &\tilde{\mathbf{X}}^\mathbf{T}\mathbf{D}_\perp^\mathbf{T}\mathbf{D}_\perp\tilde{\mathbf{X}})\\ &\geq \text{trace}(\tilde{\mathbf{Y}}^\mathbf{T}\tilde{\mathbf{P}}^\mathbf{T}\tilde{\mathbf{P}}\tilde{\mathbf{Y}} + \tilde{\mathbf{Y}}^\mathbf{T}\tilde{\mathbf{P}}^\mathbf{T}\mathbf{D}_\parallel\tilde{\mathbf{X}} + \tilde{\mathbf{X}}^\mathbf{T}\mathbf{D}_\parallel^\mathbf{T}\mathbf{D}_\parallel\tilde{\mathbf{X}}). \quad (30)\end{aligned}$$

$$\begin{aligned}&\textbf{Second Term} = \text{trace}((\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}_{\mathbf{in}})^T(\tilde{\mathbf{P}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}_{\mathbf{in}}))\\ &= \text{trace}(\tilde{\mathbf{Y}}^\mathbf{T}\tilde{\mathbf{P}}^\mathbf{T}\tilde{\mathbf{P}}\tilde{\mathbf{Y}} + \tilde{\mathbf{Y}}^\mathbf{T}\tilde{\mathbf{P}}^\mathbf{T}\mathbf{D}_\parallel\tilde{\mathbf{X}}_{\mathbf{in}} + \tilde{\mathbf{X}}_{\mathbf{in}}^\mathbf{T}\mathbf{D}_\parallel^\mathbf{T}\mathbf{D}_\parallel\tilde{\mathbf{X}}_{\mathbf{in}} + \\ &\tilde{\mathbf{X}}_{\mathbf{in}}^\mathbf{T}\mathbf{D}_\perp^\mathbf{T}\mathbf{D}_\perp\tilde{\mathbf{X}}_{\mathbf{in}})\\ &\geq \text{trace}(\tilde{\mathbf{Y}}^\mathbf{T}\tilde{\mathbf{P}}^\mathbf{T}\tilde{\mathbf{P}}\tilde{\mathbf{Y}} + \tilde{\mathbf{Y}}^\mathbf{T}\tilde{\mathbf{P}}^\mathbf{T}\mathbf{D}_\parallel\tilde{\mathbf{X}}_{\mathbf{in}} + \tilde{\mathbf{X}}_{\mathbf{in}}^\mathbf{T}\mathbf{D}_\parallel^\mathbf{T}\mathbf{D}_\parallel\tilde{\mathbf{X}}_{\mathbf{in}}). \quad (31)\end{aligned}$$

$$\begin{aligned}&\textbf{Third Term} = \text{trace}(\mathbf{D}\tilde{\mathbf{X}}_{\mathbf{out}})^\mathbf{T}(\mathbf{D}\tilde{\mathbf{X}}_{\mathbf{out}}))\\ &= \text{trace}(\tilde{\mathbf{X}}_{\mathbf{out}}^\mathbf{T}\mathbf{D}_\parallel^\mathbf{T}\mathbf{D}_\parallel\tilde{\mathbf{X}}_{\mathbf{out}} + \tilde{\mathbf{X}}_{\mathbf{out}}^\mathbf{T}\mathbf{D}_\perp^\mathbf{T}\mathbf{D}_\perp\tilde{\mathbf{X}}_{\mathbf{out}}).\\ &\geq \text{trace}(\tilde{\mathbf{X}}_{\mathbf{out}}^\mathbf{T}\mathbf{D}_\parallel^\mathbf{T}\mathbf{D}_\parallel\tilde{\mathbf{X}}_{\mathbf{out}}) \quad (32)\end{aligned}$$

The equality is reached when $\mathbf{D}_\perp = \mathbf{0}$. Hence, the form of $\mathbf{D}^*$ is:

$$\mathbf{D}^* = \tilde{\mathbf{P}}\tilde{\mathbf{Y}}\tilde{\mathbf{B}}.$$

Form for $\mathbf{P}_\mathbf{i}^*$: For each $i = 1, \cdots, M$, $\mathbf{P}_\mathbf{i}^*$ can be decomposed as:

$$\mathbf{P}_\mathbf{i}^* = \mathbf{P}_{\parallel,\mathbf{i}} + \mathbf{P}_{\perp,\mathbf{i}} \quad (33)$$

$$\text{where, } \mathbf{P}_{\parallel,\mathbf{i}} = (\mathbf{Y_i A_i})^\mathbf{T}, \mathbf{P}_{\perp,\mathbf{i}}\mathbf{Y_i} = \mathbf{0}. \quad (34)$$

Let $\tilde{\mathbf{P}}_\parallel = [\mathbf{P}_{\parallel,\mathbf{1}}, \cdots, \mathbf{P}_{\parallel,\mathbf{M}}]$ and $\tilde{\mathbf{P}}_\perp = [\mathbf{P}_{\perp,\mathbf{1}}, \cdots, \mathbf{P}_{\perp,\mathbf{M}}]$. Substituting the value for $\mathbf{D}^*$ into cost terms, we can write the terms of $\mathcal{C}_1$ as:

$$\begin{aligned}&\textbf{First Term} = \|\tilde{\mathbf{P}}^*\tilde{\mathbf{Y}}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})\|_F^2\\ &= \|(\tilde{\mathbf{P}}_\parallel + \tilde{\mathbf{P}}_\perp)\tilde{\mathbf{Y}}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})\|_F^2\\ &= \|\tilde{\mathbf{P}}_\parallel\tilde{\mathbf{Y}}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})\|_F^2\\ &= \text{trace}(\tilde{\mathbf{P}}_\parallel\tilde{\mathbf{Y}}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})^\mathbf{T}\tilde{\mathbf{Y}}^\mathbf{T}\tilde{\mathbf{P}}_\parallel^\mathbf{T}). \quad (35)\end{aligned}$$

$$\begin{aligned}&\textbf{Second Term} = \|\tilde{\mathbf{P}}^*\tilde{\mathbf{Y}}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})\|_F^2\\ &= \|(\tilde{\mathbf{P}}_\parallel + \tilde{\mathbf{P}}_\perp)\tilde{\mathbf{Y}}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})\|_F^2\\ &= \|\tilde{\mathbf{P}}_\parallel\tilde{\mathbf{Y}}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})\|_F^2\\ &= \text{trace}(\tilde{\mathbf{P}}_\parallel\tilde{\mathbf{Y}}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})^\mathbf{T}\tilde{\mathbf{Y}}^\mathbf{T}\tilde{\mathbf{P}}_\parallel^\mathbf{T}). \quad (36)\end{aligned}$$

$$\begin{aligned}&\textbf{Third Term} = \|\tilde{\mathbf{P}}^*\tilde{\mathbf{Y}}(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})\|_F^2\\ &= \|(\tilde{\mathbf{P}}_\parallel + \tilde{\mathbf{P}}_\perp)\tilde{\mathbf{Y}}(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})\|_F^2\\ &= \|\tilde{\mathbf{P}}_\parallel\tilde{\mathbf{Y}}(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})\|_F^2\\ &= \text{trace}(\tilde{\mathbf{P}}_\parallel\tilde{\mathbf{Y}}(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})^\mathbf{T}\tilde{\mathbf{Y}}^\mathbf{T}\tilde{\mathbf{P}}_\parallel^\mathbf{T}). \quad (37)\end{aligned}$$

The cost term, $\mathcal{C}_2$ can be written as:

$$\begin{aligned}&\mathcal{C}_2(\tilde{\mathbf{P}}) = -\text{trace}((\tilde{\mathbf{P}}\tilde{\mathbf{Y}})(\tilde{\mathbf{P}}\tilde{\mathbf{Y}})^T)\\ &= -\text{trace}(((\tilde{\mathbf{P}}_\parallel + \tilde{\mathbf{P}}_\perp)\tilde{\mathbf{Y}})((\tilde{\mathbf{P}}_\parallel + \tilde{\mathbf{P}}_\perp)\tilde{\mathbf{Y}})^T)\\ &= -\text{trace}((\tilde{\mathbf{P}}_\parallel\tilde{\mathbf{Y}})(\tilde{\mathbf{P}}_\parallel\tilde{\mathbf{Y}})^T). \quad (38)\end{aligned}$$

Putting all the terms together, the overall objective function becomes:

$$\begin{aligned}&\text{trace}(\tilde{\mathbf{P}}_\parallel\tilde{\mathbf{Y}}((\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})^\mathbf{T} + \mu(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})\\ &(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})^\mathbf{T} + \nu(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})^\mathbf{T} - \lambda\mathbf{I})\tilde{\mathbf{Y}}^\mathbf{T}\tilde{\mathbf{P}}_\parallel^\mathbf{T})\\ &= \text{trace}(\tilde{\mathbf{A}}_\mathbf{T}\tilde{\mathbf{K}}((\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})^\mathbf{T} + \mu(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})\\ &(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})^\mathbf{T} + \nu(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})^\mathbf{T} - \lambda\mathbf{I})\tilde{\mathbf{K}}\tilde{\mathbf{A}}). \quad (39)\end{aligned}$$

It can be seen that from equation (39), that the cost function is independent of $\mathbf{P}_{\perp,\mathbf{i}}$, hence it can be safely set to be $\mathbf{0}$. Hence,

$$\mathbf{P}_\mathbf{i}^* = (\mathbf{Y_i A_i})^\mathbf{T}.$$

1) Updating $\tilde{\mathbf{A}}$*:* Using Proposition 1, optimization problem equation (6) becomes:

$$\{\tilde{\mathbf{A}}^*, \tilde{\mathbf{B}}^*, \mathbf{X}^*\} = \arg\min_{\tilde{\mathbf{A}},\tilde{\mathbf{B}},\tilde{\mathbf{X}}} \mathcal{C}_1(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{X}}) + \lambda \mathcal{C}_2(\tilde{\mathbf{A}})$$
$$\text{s.t. } \mathbf{A_i^T K_i A_i} = \mathbf{I},\ i = 1, \cdots, M \text{ and } \|\tilde{\mathbf{x}}_j\|_1 \leq T_0, \forall j. \quad (40)$$

Here, we assume that $(\tilde{\mathbf{B}}, \tilde{\mathbf{X}})$ are fixed. Then, the optimization for $\tilde{\mathbf{A}}$ can be solved efficiently. We have the following proposition.

2) Proposition 2:: The optimal solution of equation (40) when $(\tilde{\mathbf{B}}, \tilde{\mathbf{X}})$ *are fixed is:*

$$\{\mathbf{G}^*\} = \arg\min_{\mathbf{G}} \text{trace}[\mathbf{G^T H G}]$$
$$s.t.\ \mathbf{G_i^T G_i} = \mathbf{I}\ \forall\ i\ = 1, \cdots, M \quad (41)$$

where,

$$\mathbf{H} = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^{\mathbf{T}}((\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})^{\mathbf{T}} + \mu(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})$$
$$(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})^{\mathbf{T}} + \nu(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})^{\mathbf{T}} - \lambda\mathbf{I})\mathbf{V}\mathbf{S}^{\frac{1}{2}} \quad (42)$$

Proof:
Let,

$$\tilde{\mathbf{K}} = \mathbf{V}\mathbf{S}\mathbf{V}^{\mathbf{T}},$$

$$\tilde{\mathbf{H}} = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^{\mathbf{T}}((\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}})^{\mathbf{T}} +$$
$$\mu(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}})^{\mathbf{T}} + \nu(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})(\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}})^{\mathbf{T}}$$
$$-\lambda\mathbf{I})\mathbf{V}\mathbf{S}^{\frac{1}{2}},$$

and

$$\mathbf{G} = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^{\mathbf{T}}\tilde{\mathbf{A}}.$$

Substituting into equation (39), we get the required form of the optimization.

REFERENCES

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, Nov 2006. 2

[2] M. Baktashmotlagh, M. Harandi, B. Lovell, and M. Salzmann. Unsupervised domain adaptation by domain invariant projection. In *IEEE International Conference on Computer Vision*, pages 769–776, Dec 2013. 2

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003. 3

[4] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, Nov 1986. 8

[5] M. Chen, Z. Xu, F. Sha, and K. Q. Weinberger. Marginalized denoising autoencoders for domain adaptation. In *International Conference on Machine Learning*, pages 767–774, 2012. 2

[6] S. Chopra, S. Balakrishnan, and R. Gopalan. Dlid: Deep learning for domain adaptation by interpolating between domains. In *ICML Workshop on Challenges in Representation Learning*, 2013. 2

[7] H. Daumé III. Frustratingly easy domain adaptation. In *ACL*, 2007. 1

[8] L. Duan, I. W. Tsang, and D. Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):465–479, 2012. 2

[9] L. Duan, I. W. Tsang, D. Xu, and T.-S. Chua. Domain adaptation from multiple sources via auxiliary classifiers. In *International Conference on Machine Learning*, ICML '09, pages 289–296, New York, NY, USA, 2009. ACM. 2

[10] L. Duan, D. Xu, and S.-F. Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1338–1345, June 2012. 2

[11] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, Dec 2006. 1

[12] B. Fernando, A. Habrard, M. Sebban, T. Tuytelaars, et al. Unsupervised visual domain adaptation using subspace alignment. *IEEE International Conference on Computer Vision*, 2013. 2

[13] I. Gkioulekas and T. Zickler. Dimensionality reduction using the sparse linear model. In *NIPS*, pages 271–279, 2011. 2

[14] B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In S. Dasgupta and D. Mcallester, editors, *International Conference on Machine Learning*, pages 222–230. JMLR Workshop and Conference Proceedings, 2013. 2

[15] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, June 2012. 2, 7, 8

[16] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *IEEE International Conference on Computer Vision*, 2011. 2, 7, 8

[17] R. Gopalan, R. Li, and R. Chellappa. Unsupervised adaptation across domain shift by generating intermediate data representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2014. 7, 8

[18] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. 6

[19] R. Gross, I. Matthews, J. F. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image Vision Computing*, 28(5):807–813, 2010. 6

[20] Y. Han, F. Wu, D. Tao, J. Shao, Y. Zhuang, and J. Jiang. Sparse unsupervised dimensionality reduction for multiple view data. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(10):1485–1496, Oct. 2012. 2

[21] I.-H. Jhuo, D. Liu, D. Lee, and S.-F. Chang. Robust visual domain adaptation with low-rank reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2168–2175, June 2012. 2, 7, 8

[22] Y. Jia, M. Salzmann, and T. Darrell. Factorized latent spaces with structured sparsity. In *NIPS*, pages 982–990, 2010. 1

[23] E. Kokiopoulou, J. Chen, and Y. Saad. Trace optimization and eigenproblems in dimension reduction methods. *Numerical Linear Algebra with Applications*, 18(3):565–602, 2011. 3

[24] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1785–1792, June 2011. 2, 8, 9, 10

[25] W. Li, L. Duan, D. Xu, and I. Tsang. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1134–1148, June 2014. 2, 7, 8, 10

[26] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):791–804, April 2012. 2

[27] V. Monga, N. Damera-Venkata, H. Rehman, and B. Evans. Halftoning matlab toolbox, 2005. 8

[28] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa. Sparse embedding: A framework for sparsity promoting dimensionality reduction. In *European Conference on Computer Vision*, pages 414–427, Oct. 2012. 2, 5

[29] J. Ni, Q. Qiu, and R. Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 692–699, June 2013. 1, 7, 8

[30] B. Olshausen and D. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997. 1

[31] Y. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Asilomar Conference on Signals, Systems and Computers*, 1993. 5

[32] J. Pillai, A. Shrivastava, V. Patel, and R. Chellappa. Learning discriminative dictionaries with partially labeled data. In *IEEE Conference on Image Processing*, Oct. 2012. 3

[33] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa. Domain adaptive dictionary learning. In *ECCV*, 2012. 1

[34] I. Ramírez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3501–3508, June 2010. 3

[35] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category

models to new domains. In *ECCV*, volume 6314, pages 213–226, 2010. 2, 6, 7, 8, 10

[36] L. K. Saul and S. T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, Dec 2003. 3

[37] A. Sharma and D. W. Jacobs. Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, June 2011. 6

[38] A. Sharma, A. Kumar, H. D. III, and D. W. Jacobs. Generalized multiview analysis: A discriminative latent space. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2160–2167, June 2012. 6

[39] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa. Generalized domain-adaptive dictionaries. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 361–368, 2013. 2

[40] X. Shi, Q. Liu, W. Fan, P. Yu, and R. Zhu. Transfer learning on heterogenous feature spaces via spectral transformation. In *International Conference on Data Mining*, pages 1049–1054, Dec 2010. 8, 10

[41] Y. Shi and F. Sha. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 1079–1086, 2012. 2

[42] A. Shrivastava, S. Shekhar, and V. M. Patel. Unsupervised domain adaptation using parallel transport on grassmann manifold. In *IEEE Winter conference on Applications of Computer Vision*, 2014. 2

[43] C. Wang and S. Mahadevan. Heterogeneous domain adaptation using manifold alignment. In *International Joint Conference on Artificial Intelligence*, pages 1541–1546. AAAI Press, 2011. 8, 10

[44] S. Wang, L. Zhang, Y. Liang, and Q. Pan. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2216–2223, June 2012. 2

[45] Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. Technical report, Rice University, 2010. 4, 5

[46] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009. 1

[47] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang. Coupled dictionary training for image super-resolution. *IEEE Transactions on Image Processing*, 21(8):3467–3478, Aug. 2012. 2

[48] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive SVMs. In *International Conference on Multimedia*, pages 188–197. ACM, 2007. 2

[49] M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher Discrimination Dictionary learning for sparse representation. In *IEEE International Conference on Computer Vision*, pages 543–550, Nov. 2011. 3, 4, 5, 6, 7, 8

[50] L. Zhang, M. Yang, Z. Feng, and D. Zhang. On the dimensionality reduction for sparse representation based face recognition. In *International Conference on Pattern Recognition*, pages 1237–1240, Aug. 2010. 2

[51] J. T. Zhou, I. W. Tsang, S. J. Pan, and M. Tan. Heterogeneous domain adaptation for multiple classes. In *International Conference on Artificial Intelligence and Statistics*, pages 1095–1103, 2014. 2