ABSTRACT

Title of dissertation: SPARSE DICTIONARY LEARNING AND DOMAIN ADAPTATION FOR FACE AND ACTION RECOGNITION

Qiang Qiu, Doctor of Philosophy, 2013

Dissertation directed by: Professor Rama Chellappa
Department of Computer Science

New approaches for dictionary learning and domain adaptation are proposed for face and action recognition. We first present an approach for dictionary learning of action attributes via information maximization. We unify the class distribution and appearance information into an objective function for learning a sparse dictionary of action attributes. The objective function maximizes the mutual information between what has been learned and what remains to be learned in terms of appearance information and class distribution for each dictionary atom. We propose a Gaussian Process (GP) model for sparse representation to optimize the dictionary objective function. Hence we can describe an action video by a set of compact and discriminative action attributes. More importantly, we can recognize modeled action categories in a sparse feature space, which can be generalized to unseen and unmodeled action categories.

We then extend the attribute-based approach to a two-stage information-driven dictionary learning framework for general image classification tasks. The proposed method seeks a dictionary that is compact, discriminative, and generative. In the first stage, dictio-

nary atoms are selected from an initial dictionary by maximizing the mutual information measure on dictionary compactness, discrimination and reconstruction. In the second stage, the selected dictionary atoms are updated for improved reconstructive and discriminative power using a simple gradient ascent algorithm on mutual information.

When designing dictionaries, training and testing domains may often be different, due to different view points and illumination conditions. We further present a domain adaptive dictionary learning framework for the task of transforming a dictionary learned from one visual domain to the other, while maintaining a domain-invariant sparse representation of a signal. Domain dictionaries are modeled by a linear or non-linear parametric function. The dictionary function parameters and domain-invariant sparse codes are then jointly learned by solving an optimization problem.

Finally, in the context of face recognition, we present a dictionary learning approach to compensate for the transformation of faces due to changes in view point, illumination, resolution, etc. The approach is to first learn a domain base dictionary, and then describe each domain shift (identity, pose, illumination) using a sparse representation over the base dictionary. The dictionary adapted to each domain is expressed as sparse linear combinations of the base dictionary. With the proposed compositional dictionary approach, a face image can be decomposed into sparse representations for a given subject, pose and illumination respectively. The extracted sparse representation for a subject is consistent across domains and enables pose and illumination insensitive face recognition. Sparse representations for pose and illumination can be used to estimate the pose and illumination condition of a face image. By composing sparse representations for subjects and domains, we can also perform pose alignment and illumination normalization.

SPARSE DICTIONARY LEARNING AND DOMAIN ADAPTATION FOR FACE AND ACTION RECOGNITION

by

Qiang Qiu

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2013

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor John Benedetto
Professor Larry Davis
Professor Amol Deshpande
Professor Amitabh Varshney

Dedication

To my family.

Acknowledgments

First and foremost I would like to thank my advisor, Professor Rama Chellappa for accepting me as his student, and supporting me to work on this challenging and interesting topic over the past three years. He has always made himself available whenever I sought his advices. The discussions with him were always encouraging and inspiring. His dedication to work, positive attitude towards life, and polite personality will all remain an inspiration for me in my future career.

I would like to thank Professor Amol Deshpande for mentoring me in the first two years of my PhD study, and serving on both my proposal and dissertation committee. It has been a precious experience to work with and learn from him. I would like to also thank Professor Larry Davis for valuable guidance on research projects and thesis. Thanks are due to Professor John Benedetto and Professor Amitabh Varshney for agreeing to serve on my dissertation committee and for sparing their invaluable time reviewing the manuscript.

My graduate life has been enriched in many ways by fellow colleagues at the Computer Vision Lab, among whom I should particularly mention Juncheng Chen, Ming Du, Qi Hu, Zhuolin Jiang, Mingyu Liu, Jie Ni, Vishal Patel, Sima Taheri and Pavan Turaga for their fruitful discussion and collaboration on research projects. I would also like to acknowledge administrative help from Ms. Janice Perrone.

I owe my deepest thanks to Zhengyi. This work will be impossible without you.

It is impossible to remember all, and I apologize to those I've inadvertently left out.

Table of Contents

List of Figures

Chapter 1

Introduction

Describing human actions and faces using attributes is closely related to representing an object using attributes [10]. Several studies have investigated the attribute-based approaches for object recognition problems [10–14]. These methods have demonstrated that attribute-based approaches can not only recognize object categories, but can also describe unknown object categories. In this dissertation, we first present a dictionary-based approach for learning human action attributes which are useful to model and recognize known action categories, and also describe unknown action categories. We then extend the action attributes learning approach to an information-theoretic dictionary learning framework for general image classification tasks. When designing dictionaries, we often face the problem that training and testing domains may be different, due to different view points and illumination conditions. We further propose a domain adaptive dictionary learning framework for the task of transforming a dictionary learned from one visual domain to the other, while maintaining a domain-invariant sparse representation of a signal. Finally, we discuss a compositional dictionary approach for domain adaptive face recognition. The dictionary adapted to each domain is expressed as sparse linear combinations of a base dictionary.

1.1 Sparse Dictionary-based Attributes Learning

In the first contribution, we consider dictionary learning of human action attributes through information maximization. In addition to using the appearance information between dictionary atoms, we also exploit the class label information associated with dictionary atoms to learn a compact and discriminative dictionary for human action attributes. The mutual information for appearance information and class distributions between the learned dictionary and the rest of the dictionary space are used to define the objective function, which is optimized using a Gaussian Process (GP) model [15] proposed for sparse representation. The property of sparse coding naturally leads to a GP kernel with compact support resulting in significant speed-ups. The representation and recognition of actions are through sparse coefficients related to learned attributes. A compact and discriminative attribute dictionary should encourage the signals from the same class to have very similar sparse representations. In other words, the signals from the same class are described by a similar set of dictionary atoms with similar coefficients, which is critical for classification using learned dictionaries. Experimental results on four public action datasets demonstrate the effectiveness of our approach in action recognition and summarization.

1.2 Information-theoretic Dictionary Learning

In the second contribution, we extend the action attributes learning approach to a two-stage information-theoretic dictionary learning framework for general image classification tasks. A key feature of our framework is that it can learn not only reconstructive but also compact and discriminative dictionaries. Our method consists of two main stages

involving greedy atom selection and simple gradient ascent atom updates, resulting in a highly efficient algorithm. In the first stage, dictionary atoms are selected in a greedy way such that the common internal structure of signals belonging to a certain class is extracted while simultaneously maintaining global discrimination among the classes. In the second stage, the dictionary is updated for better discrimination and reconstruction via a simple gradient ascent method that maximizes the mutual information (MI) between the signals and the dictionary, as well as the sparse coefficients and the class labels. Experiments using public object and face datasets demonstrate the effectiveness of our approach for image classification tasks.

1.3 Domain Adaptive Dictionary Learning

In the third contribution, we explore a function learning framework for the task of transforming a dictionary learned from one visual domain to the other, while maintaining a domain-invariant sparse representation of a signal. When designing dictionaries for image classification tasks, we are often confronted with situations where conditions, e.g., view points and illumination, in the training set are different from those present during testing. Given the same set of signals observed in different visual domains, our goal is to learn a dictionary for the new domain without corresponding observations. We formulate this problem of dictionary transformation in a function learning framework, i.e., dictionaries across different domains are modeled by a parametric function. The dictionary function parameters and domain-invariant sparse codes are then jointly learned by solving an optimization problem. The problem of transforming a dictionary trained from one

visual domain to another without changing signal sparse representations can be viewed as a problem of domain adaptation [16] and transfer learning [17]. We demonstrate the effectiveness of our approach for applications such as face recognition, pose alignment and pose estimation.

1.4 Domain Adaptive Compositional Dictionary Learning

In the final contribution, we present a compositional dictionary approach for domain adaptive face recognition. Face recognition across domains, e.g., pose and illumination, has proved to be a challenging problem [6,18,19]. We propose to first learn a domain base dictionary, and then describe each domain shift (identity, pose, illumination) using a sparse representation over the base dictionary. The dictionary adapted to each domain is then expressed as sparse linear combinations of the base dictionary. Using this approach, a face image can be decomposed into sparse representations for a given subject, pose and illumination respectively. This approach has three advantages: first, the extracted sparse representation for a subject is consistent across domains and enables pose and illumination insensitive face recognition. Second, sparse representations for pose and illumination can subsequently be used to estimate the pose and illumination condition of a face image. Finally, by composing sparse representations for subject and the different domains, we can also perform pose alignment and illumination normalization. Extensive experiments using two public face datasets are presented to demonstrate the effectiveness of our approach for face recognition across domains.

1.5 Organization of the Dissertation

In Chapter 2, we discuss an approach for dictionary learning of action attributes via information maximization. In Chapter 3, we introduce a two-stage information-theoretic dictionary learning framework for image classification tasks. In Chapter 4, a domain adaptive dictionary learning framework is presented. In Chapter 5, a compositional dictionary approach is discussed for domain adaptive face recognition. Finally, in Chapter 6, we discuss directions for future work.

Chapter 2

Dictionary-based Attributes for Action Recognition and Summarization

2.1 Introduction

Dictionary learning is one of the approaches for learning attributes (i.e., dictionary atoms) from a set of training samples. In [20], a promising dictionary learning algorithm, K-SVD, is introduced to learn an over-complete dictionary. Input signals can then be represented as a sparse linear combination of dictionary atoms. K-SVD only focuses on focus on representational capability, i.e., minimizes the reconstruction error. The method of optimal direction (MOD) [21] shares the same sparse coding as K-SVD. [22] manually selects training samples to construct a dictionary. [23] trains one dictionary for each class to obtain discriminability.

Discriminative dictionary learning is gaining attention in many disciplines. Discriminative K-SVD in [24] extends K-SVD by incorporating the classification error into the objective function to obtain a more discriminative dictionary. [25] aims to obtain the discriminative power of dictionary by iteratively updating the dictionary from the results of a linear classifier. [26] introduces a label consistent constraint to obtain the discrimination of sparse codes among the classes. Some other examples include LDA-based basis selection [27], distance matrix learning [28], hierarchical pairwise merging of visual words [29], maximization of mutual information (MMI) [1, 30, 31], and sparse coding-based dictionary learning [23, 32].

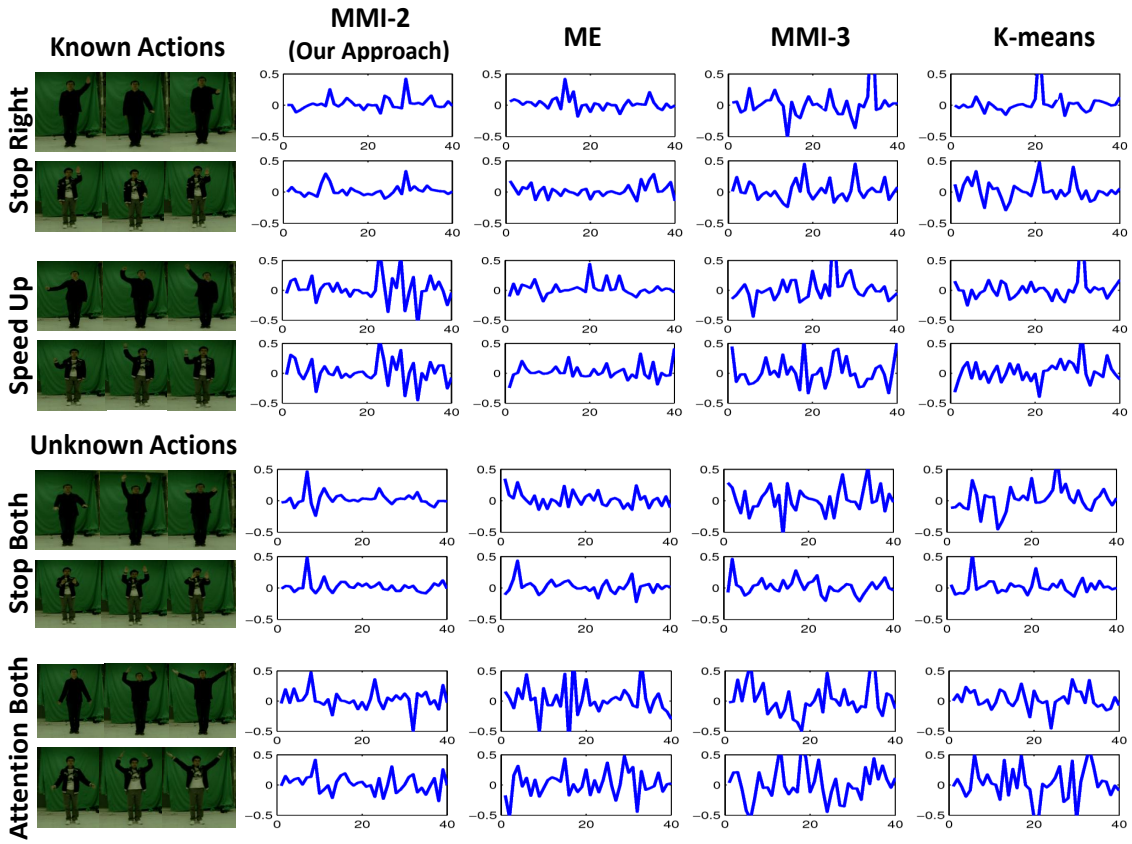


Figure 2.1: Sparse representations of four actions (two are known and two are unknown to the attribute dictionary) using attribute dictionaries learned by different methods. Each action is performed by two different humans. For visualization purpose, each waveform shows the average of the sparse codes of all frames in an action sequence. We learned several attribute dictionaries using methods including our approach, the Maximization of Entropy approach (ME), the MMI-3 approach motivated by [1] and the K-means approach. A compact and discriminative attribute dictionary should encourage actions from the same class to be described by a similar set of attributes, i.e., similar sparse codes. The attribute dictionary learned by our approach provides similar waveforms, which shows consistent sparse representations, for the same class action sequences.

Recent dictionary-based approaches for learning action attributes include agglomerative clustering [33], forward selection [34] and probabilistic graphical model [35]. [36] proposes an unsupervised approach and uses L_1 minimization to find basic primitives to represent human motions.

In this chapter, we propose an approach for dictionary learning of human action attributes via information maximization. In addition to using the appearance information between dictionary atoms, we also exploit class label information associated with dictionary atoms to learn a compact and discriminative dictionary for human action attributes. The mutual information for appearance information and class distributions between the learned dictionary and the rest of the dictionary space are used to define the objective function, which is optimized using a Gaussian Process (GP) model [15] proposed for sparse representation. The property of sparse coding naturally leads to a kernel with compact support, i.e., zero values for a most portion, in GP for significant speed-ups. Representation and recognition of actions are accomplished through sparse coefficients related to learned attributes.

Unlike previous dictionary learning methods that mostly consider learning reconstructive dictionaries, our algorithm can encourage dictionary compactness and discriminability simultaneously. Sparse representation over a dictionary with coherent atoms has the multiple representation problem [37]. A compact dictionary consists of incoherent atoms, and encourages similar signals, which are more likely from the same class, to be consistently described by a similar set of atoms with similar coefficients. A discriminative dictionary encourages signals from different classes to be described by either a different set of atoms, or the same set of atoms but with different coefficients [23, 37, 38].

Both aspects are critical for action classification using sparse representation. As shown in Fig. 2.1, our approach produces consistent sparse representations for the same class of signals.

Our approach adopts the rule of Maximization of Mutual Information to obtain a compact and discriminative dictionary. The dictionary atoms are considered as attributes in our approach. Compared to previous methods, our approach maximizes the mutual information for both the appearance information and class distribution of dictionary atoms to learn a dictionary while [31] and [1] only maximize the mutual information for class distribution. Thus, we can expect improved dictionary compactness from our approach. Both [31] and [1] obtain a dictionary through merging of two visual words, which can be time-consuming when the dictionary size is large. Besides, our approach is efficient because the dictionary is learned in the sparse feature space so we can leverage the property of sparse coding to use kernel locality for speeding up the dictionary learning process.

This chapter makes the following contributions:

- We propose a novel probabilistic model for sparse representation.
- We learn a compact and discriminative dictionary for sparse coding via information maximization.
- We describe and recognize human actions, including unknown actions, via a set of human action attributes in a sparse feature space.
- We present a simple yet near-optimal action summarization method.

The rest of this chapter is structured as follows. In Sec. 2.2, we discuss human action features and attributes. We then propose a novel probabilistic model for sparse

representation in Sec. 2.3. In Sec. 2.4, we present our attribution dictionary learning framework. We describe how to adopt our attribution dictionary learning method for action summarization in Sec. 2.5. Experimental results are given in Sec. 2.6 to demonstrate the effectiveness of our approach for action recognition and summarization.

2.2 Action Features and Attributes

Human action features are extracted from an action interest region for representing and describing actions. The action interest region is defined as a bounded region around the human performing the activity, which is obtained using background subtraction and/or tracking.

2.2.1 Basic Features

The human action attributes require feature descriptors to represent visual aspects. We introduce basic features, including both local and global features, used in the chapter.

Global Features: Global features encode rich information from an action interest region, so they generally perform better than local features in recognition. When cameras and backgrounds are static, we use the silhouette-based feature descriptor presented in [39] to capture shape information, while we use Histogram of oriented gradient (HOG) descriptors used in [40] for dynamic backgrounds and moving cameras. For encoding motion information, we use optical-flow based feature descriptors as in [41]. We use Action Bank descriptors introduced in [42] to demonstrate that our attribute learning method can enhance the discriminability of high-level global features.

Local Features: Spatio-temporal local features describe a video as a collection of independent patches or 3D cuboids, which are less sensitive to viewpoint changes, noise and partial occlusion. We first extract a collection of space-time interest points (STIP) introduced in [2] to represent an action sequence, and then use HOG and histogram of flow to describe them.

2.2.2 Human Action Attributes

Motivated by [33–35], an action can be represented as a set of basic action units. We refer to these basic action units as human action attributes. In order to effectively describe human actions, we need to learn a representative and semantic set of action attributes. Given all the basic features from training data, we aim to learn a compact and discriminative dictionary where all the dictionary atoms can be used as human action attributes. The final learned dictionary can be used as a "Thesaurus" of human action attributes. Each human action is then decomposed as sparse linear combinations of attributes in the thesaurus though sparse coding. The sparse coefficient associated with each attribute measures its weight in representing an action.

2.3 A Probabilistic Model for Sparse Representation

Before we present our dictionary learning framework, we first suggest a novel probabilistic model for sparse representation motivated by [43].

2.3.1 Reconstructive Dictionary Learning

A reconstructive dictionary can be learned through K-SVD [20], which is a method to learn an over-complete dictionary for sparse coding. Let Y be a set of N input signals in a n-dimensional feature space $Y = [y_1 ... y_N],\ y_i \in \mathbb{R}^n$. In K-SVD, a dictionary with a fixed number of K atoms is learned by finding a solution iteratively to the following problem:

$$\arg\min_{D,X} \|Y - DX\|_2^2 \qquad s.t. \forall i, \|x_i\|_0 \leq T \tag{2.1}$$

where $D = [d_1 ... d_K],\ d_i \in \mathbb{R}^n$ $(K > n)$ is the learned dictionary, $X = [x_1, ..., x_N],\ x_i \in \mathbb{R}^K$ are the sparse codes of input signals Y, and T specifies the sparsity that each signal has fewer than T atoms in its decomposition. Each dictionary atom d_i is L_2-normalized. The learned dictionary D from (2.1) only minimizes the reconstruction error, so it is not optimal in terms of compactness and discriminability.

2.3.2 A Gaussian Process

Given a set of input signals Y, $Y = [y_1 ... y_N], y_i \in \mathbb{R}^n$, there exists an infinite dictionary space $\mathcal{D} \subseteq \mathbb{R}^n$. Each dictionary atom $d_i \in \mathcal{D}$ maps the set of input signals to its corresponding sparse coefficients $x_{d_i} = [x_{i,1} ... x_{i,N}]$ in X, which can be viewed as its observations to the set of input signals. When two dictionary atoms d_i and d_j are similar, it is more likely that input signals will use them simultaneously in their sparse decomposition [13]. Thus the similarity of two dictionary atoms can be assessed by the correlation between their observations (i.e., sparse coefficients). Such correlation property of sparse coefficients has been used in [13] to cluster dictionary atoms.

With the above formulation, we obtain a problem which is commonly referred as a GP model. A GP is specified by a mean function and a symmetric positive-definite covariance function $\mathcal{K}$. Since we simplify our problem by assuming an initial dictionary D^o, we only need to specify entries in the covariance function $\mathcal{K}$ for atoms existing in D^o, and leave the rest undefined. For each pair of dictionary atoms $\forall d_i, d_j \in D^o$, the corresponding covariance function entry $\mathcal{K}(i,j)$ is defined as the covariance between their associated sparse coefficients $cov(x_{d_i}, x_{d_j})$. For simplicity, we use the notation $\mathcal{K}_{(d_i,d_j)}$ to refer to the covariance entry at the indices of d_i, d_j. Similarly, we use $\mathcal{K}_{(D^*,D^*)}$ to denote the covariance matrix for a set of dictionary atoms D^*.

The GP model for sparse representation provides the following useful property: given a set of dictionary atoms D^* and the associated sparse coefficients X_{D^*}, the distribution $P(X_{d^*}|X_{D^*})$ at any given testing dictionary atom d^* is a Gaussian with a closed-form conditional variance [15].

$$\mathbb{V}(d^*|D^*) = \mathcal{K}_{(d^*,d^*)} - \mathcal{K}^T_{(d^*,D^*)}\mathcal{K}^{-1}_{(D^*,D^*)}\mathcal{K}_{(d^*,D^*)} \tag{2.2}$$

where $\mathcal{K}_{(d^*,D^*)}$ is the vector of covariances between d^* and each atom in D^*.

2.3.3 Dictionary Class Distribution

When the set of input signals Y is labeled with one of M discrete class labels, we can further derive class related distributions over sparse representations.

As mentioned, each dictionary atom d_i maps the set of input signals to its corresponding sparse coefficients $x_{d_i} = [x_{i,1}...x_{i,N}]$ in X. Since each coefficient $x_{i,j}$ here corresponds to an input signal y_j, it is associated with a class label. If we aggregate x_{d_i}

based on class labels, we obtain a M sized vector. After normalization, we have the conditional probability $P(L|d_i),\ L \in [1, M]$, where $P(L|d_i)$ represents the probability of observing a class given a dictionary atom.

2.4 Learning Attribute Dictionary

As the optimal dictionary size is rarely known in advance, we first obtain through K-SVD an initial dictionary D^o of a large size K. As discussed, the initial dictionary D^o from (2.1) only minimizes the reconstruction error, and is not optimal in terms of compactness and discriminability. Then we learn a compact and discriminative dictionary from the initial dictionary via information maximization.

Given the initial dictionary D^o obtained from (2.1), we aim to compress it into a dictionary D^* of size k, which encourages the signals from the same class to have very similar sparse representations, as shown in Fig. 2.1. In other words, the signals from the same class are described by a similar set of attributes, i.e., dictionary atoms. Therefore, a compact and discriminative dictionary is more desirable.

An intuitive heuristic is to start with $D^* = \emptyset$, and iteratively choose the next best atom d^* from $D^o \backslash D^*$ which provides a maximum increase for the entropy of D^*, i.e., $\arg\max_{d^*} H(d^*|D^*)$, until $|D^*| = k$, where $D^o \backslash D^*$ denotes the remaining dictionary atoms after D^* have been removed from the initial dictionary D^o. Using the GP model, we can evaluate $H(d^*|D^*)$ as a closed-form Gaussian conditional entropy,

$$H(d^*|D^*) = \frac{1}{2} log(2\pi e \mathbb{V}(d^*|D^*)) \tag{2.3}$$

where $\mathbb{V}(d^*|D^*)$ is defined in (2.2). This heuristic is a good approximation to the *maxi-*

mization of joint entropy (ME) criteria, i.e., $\arg\max_{D^*} H(D^*)$.

With the ME rule, as atoms in the learned dictionary are less correlated to each other due to their high joint entropy, the learned dictionary is compact. However, the maximal entropy criteria will favor attributes associated with the beginning and the end of an action, as they are least correlated. Such a phenomenon is shown in Fig. 2.3b and Fig. 2.3d in the experiment section. Thus we expect high reconstruction error and weak discriminability. To mitigate this in our dictionary learning framework, we adopt Maximization of Mutual Information (MMI) as the criteria for ensuring dictionary compactness and discriminability.

2.4.1 MMI for Unsupervised Learning (MMI-1)

The rule of maximization of entropy only considers the entropy of dictionary atoms. Instead we choose to learn D^* that most reduces the entropy about the rest of dictionary atoms $D^o \backslash D^*$.

$$\arg\max_{D^*} I(D^*; D^o \backslash D^*) \tag{2.4}$$

It is known that maximizing the above criteria is NP-complete. A similar problem has been studied in the machine learning literature [43]. We can use a very simple greedy algorithm here. We start with $D^* = \emptyset$, and iteratively choose the next best dictionary atom d^* from $D^o \backslash D^*$ which provides a maximum increase in mutual information, i.e.,

$$\begin{aligned} \arg\max_{d^* \in D^o \backslash D^*} \quad & I(D^* \cup d^*; D^o \backslash (D^* \cup d^*)) - I(D^*; D^o \backslash D^*) \\ & = H(d^*|D^*) - H(d^*|\bar{D}^*); \end{aligned} \tag{2.5}$$

where $\bar{D}^*$ denotes $D^o \backslash (D^* \cup d^*)$. Intuitively, the ME criteria only considers $H(d^*|D^*)$, i.e., forces d^* to be most different from already selected dictionary atoms D^*, now we also consider $-H(d^*|\bar{D}^*)$ to force d^* to be most representative among the remaining atoms.

It has been proved in [43] that the above greedy algorithm is submodular and serves a polynomial-time approximation that is within $(1 - 1/e)$ of the optimum. Using arguments similar to the ones presented in [43], the near-optimality of our approach can be guaranteed if the initial dictionary size $|D^o|$ is sufficiently larger than $2|D^*|$.

Using the proposed GP model, the objective function in (2.5) can be written in a closed form using (2.2) and (2.3).

$$\arg \max_{d^* \in D^o \backslash D^*} \frac{\mathcal{K}_{(d^*,d^*)} - \mathcal{K}^T_{(d^*,D^*)} \mathcal{K}^{-1}_{(D^*,D^*)} \mathcal{K}_{(d^*,D^*)}}{\mathcal{K}_{(d^*,d^*)} - \mathcal{K}^T_{(d^*,\bar{D}^*)} \mathcal{K}^{-1}_{(\bar{D}^*,\bar{D}^*)} \mathcal{K}_{(d^*,\bar{D}^*)}} \tag{2.6}$$

Given the initial dictionary size $|D^o| = K$, each iteration requires $\mathcal{O}(K^4)$ to evaluate (2.6). Such an algorithm seems to be computationally infeasible for any large initial dictionary size. The nice feature of this approach is that we model the covariance kernel $\mathcal{K}$ over sparse codes X, which entitles $\mathcal{K}$ a compact support, i.e., most entries of $\mathcal{K}$ have zero or very tiny values. After we ignore those zero value portion while evaluating (2.6), the actual computation becomes very efficient.

2.4.2 MMI for Supervised Learning (MMI-2)

The objective functions in (2.4) and (2.5) only consider the appearance information of dictionary atoms, hence D^* is not optimized for classification. For example, attributes to distinguish a particular class can possibly be missing in D^*. So we need to use appearance

information and class distribution to construct a dictionary that also causes minimal loss information about labels.

Let L denote the labels of M discrete values, $L \in [1, M]$. In Sec. 2.3.3, we discussed how to obtain $P(L|d^*)$, which represents the probability of observing a class given a dictionary atom. Give a set of dictionary atom D^*, we define $P(L|D^*) = \frac{1}{|D^*|}\sum_{d_i \in D^*} P(L|d_i)$. For simplicity, we denote $P(L|d^*)$ as $P(L_{d^*})$, and $P(L|D^*)$ as $P(L_{D^*})$.

To enhance the discriminative power of the learned dictionary, we propose to modify the objection function (2.4) to

$$\arg\max_{D^*} I(D^*; D^o \backslash D^*) + \lambda I(L_{D^*}; L_{D^o \backslash D^*}) \tag{2.7}$$

where $\lambda \geq 0$ is the parameter to regularize the emphasis on appearance or label information. When we write (2.7) in its approximation version as (2.8)

$$\begin{aligned} \arg \max_{d^* \in D^o \backslash D^*} & [H(d^*|D^*) - H(d^*|\bar{D}^*)] \\ & + \lambda [H(L_{d^*}|L_{D^*}) - H(L_{d^*}|L_{\bar{D}^*})] \end{aligned} \tag{2.8}$$

where

$$H(L_{d^*}|L_{D^*}) = -\sum_{L \in [1,M]} P(L_{d*}) P(L_{D^*}) \log P(L_{d^*})$$

we can easily notice that now we also force the classes associated with d^* to be most different from classes already covered by selected atoms D^*; and at the same time, the classes associated with d^* should be most representative among classes covered by the remaining atoms. Thus the learned dictionary is not only compact, but also covers all classes to maintain the discriminability. It is interesting to note that MMI-1 is a special case of MMI-2 with $\lambda = 0$.

The parameters λ in (2.8) are data dependent and can be estimated as the ratio between the maximal information gained from an atom to the respective compactness and discrimination measure, i.e.,

$$\lambda = \frac{\max_{d^* \in D^o}[H(L_{d^*}|L_{D^*}) - H(L_{d^*}|L_{\bar{D}^*})]}{\max_{d^* \in D^o}[H(d^*|D^*) - H(d^*|\bar{D}^*)]}. \tag{2.9}$$

For each term in (2.8), only the first greedily selected atoms are involved in parameter estimation. This leads to an efficient process in finding the parameters.

2.4.3 MMI using dictionary class distribution (MMI-3)

MMI-1 considers the appearance information for dictionary compactness, and MMI-2 uses appearance and class distribution to enforce both dictionary compactness and discriminability. To complete the discussion, MMI-3, which is motivated by [1], only considers the dictionary class distribution, discussed in Sec. 2.3.3, for dictionary discriminability.

In MMI-3, we start with an initial dictionary D^o obtained from K-SVD. At each iteration, for each pair of dictionary atoms, d_1 and d_2, we compute the MI loss if we merge these two into a new dictionary atom d^*, and pick the pair which gives the minimum MI loss. We continue the merging process till the desired dictionary size. The MI loss is defined as,

$$\begin{aligned} \triangle I(d_1, d_2) \quad = & \sum_{L \in [1,M], i=1,2} p(d_i) p(L|d_i) \log p(L|d_i) \\ & - p(d_i) p(L|d_i) \log p(L|d^*) \end{aligned} \tag{2.10}$$

where

$$p(L|d^*) = \frac{p(d_1)}{p(d^*)}p(L|d_1) + \frac{p(d_2)}{p(d^*)}p(L|d_2)$$

$$p(d^*) = p(d_1) + p(d_2)$$

2.5 Action Summarization using MMI-1

Summarizing an action video sequence often considers two criteria: *diversity* and *coverage* [44]. The diversity criterion requires the elements in a summary be as different from each other as possible; and the coverage criterion requires a summary to also represent the original video well.

In (2.5), the first term $H(d^*|D^*)$ forces d^* to be most different from already selected dictionary atoms D^*. The second term $-H(d^*|\bar{D}^*)$ to force d^* to be most representative among the remaining atoms. By considering an action sequence as a dictionary, and each frame as a dictionary atom, MMI-1 serves a near-optimal video summarization scheme. The first term in (2.5) measures diversity and the second term in (2.5) measures coverage. The only revision required here is to define the kernel of the Gaussian process discussed in Sec. 2.3.2 as $\mathcal{K}_{(d_i,d_j)} = d_i^T d_j$.

The advantage in adopting MMI-1 as a summarization/sampling scheme can be summarized as follows: first, MMI-1 is a simple greedy algorithm that can be executed very efficiently. Second, the MMI-1 provides near-optimal sampling/summarization results, which is within $(1-1/e)$ of the optimum. Such near-optimality is achieved through a submodular objective function that enforces diversity and coverage simultaneously.

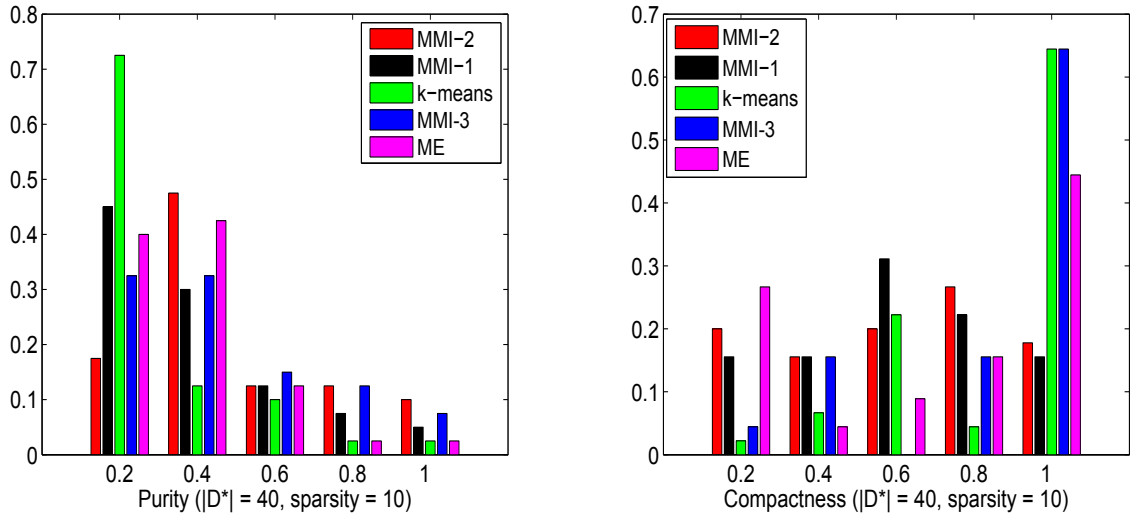
2.6 Experimental Evaluation

This section presents an experimental evaluation using four public action datasets: Keck gesture dataset [39], Weizmann action dataset [45], UCF sports action dataset [46], and UCF50 action dataset [47]. On the Keck gesture dataset, we thoroughly evaluate the basic behavior of our proposed dictionary learning approaches MMI-1, MMI-2, and MMI-3, in terms of dictionary compactness and discriminability, by comparing with other alternatives. Then we further evaluate the discriminability of our learned action attributes over the popular Weizmann aciton dataset, the challenging UCF sports and UCF50 action datasets.

2.6.1 Comparison with Alternative Approaches

The Keck gesture dataset consists of 14 different gestures, which are a subset of the military signals. These 14 classes include turn left, turn right, attention left, attention right, flap, stop left, stop right, stop both, attention both, start, go back, close distance, speed up, come near. Each of the 14 gestures is performed by three subjects. Some sample frames from this dataset are shown in Fig. 2.1.

For comparison purposes, in addition to MMI-1, MMI-2 and MMI-3 methods proposed in Sec. 2.4, we also implemented two additional action attributes learning approaches. The first approach is the maximization of entropy (ME) method discussed before. The second approach is to simply perform k-means over an initial dictionary D^o from K-SVD to obtain a desired size dictionary.



(a) Purity

(b) Compactness

Figure 2.2: Purity and compactness of learned dictionary D^*: purity is the histograms of the maximum probability observing a class given a dictionary atom, and compactness is the histograms of $D^{*T}D^*$. At the right-most bin of the respective figures, a discriminative and compact dictionary should exhibit high purity and small compactness. MMI-2 dictionary is most "pure" and second most compact (MMI-1 is most compact but much less pure.)

2.6.1.1 Dictionary Purity and Compactness

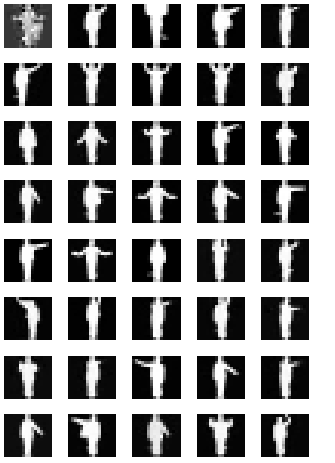
Through K-SVD, we start with an initial 500 size dictionary using the shape feature (sparsity 30 is used). We then learned a 40 size dictionary D^* from D^o using 5 different approaches. We let $\lambda = 1$ in (2.8) throughout the experiment. To evaluate the discriminability and compactness of these learned dictionaries, we evaluate the *purity* and *compactness* measures as shown in Fig. 2.2. The purity is assessed by the histograms of the maximum probability observing a class given a dictionary atom, i.e., $max(P(L|d_i))$,

and the compactness is assessed by the histograms of $D^{*T}D^{*}$. As each dictionary atom is L_2-normalized, $d_i^t d_j \in [0, 1]$ and indicates the similarity between dictionary atoms d_i and d_j. Fig. 2.2a shows MMI-2 is most "pure", as around 25% of dictionary atoms learned by MMI-2 have 0.6-above probability to only associate with one of the classes. MMI-3 shows comparable purity to MMI-2 as the MI loss criteria used in MMI-3 does retain the class information during dictionary learning. However, as shown in Fig. 2.2b, MMI-2 dictionary is much more compact, as only about 20% MMI-2 dictionary atoms have 0.80-above similarity. As expected, comparing to MMI-2, MMI-1 shows better compactness but much less purity.

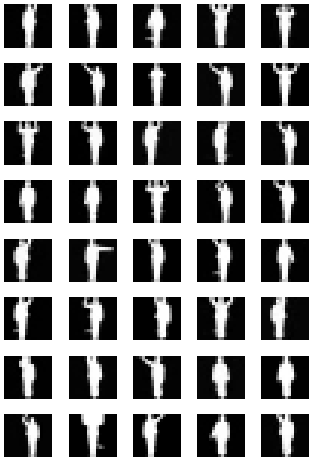
2.6.1.2 Describing Unknown Actions

We illustrate here how unknown actions can be described through a learned attribute dictionary. We first obtain a 500 size initial shape dictionary D^o using 11 out of 14 gesture classes, and keep *flap*, *stop both* and *attention both* as unknown actions. We would expect a near perfect description to these unknown actions, as we notice these three classes are composed by attributes observed in the rest classes. For example, *flap* is a two-arm gesture "unseen" by the attribute dictionary, but its left-arm pattern is similar to *turn left*, and right-arm is similar to *turn right*.

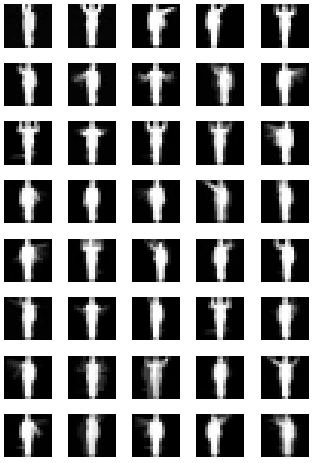
As shown in Fig. 2.3, we learned 40 size dictionaries using MMI-2, ME and MMI-3 respectively from D^o. Through visual observation, ME dictionary (Fig. 2.3b) is most compact as dictionary atoms look less similar to each other. However, different from MMI-2 dictionary (Fig. 2.3a), it contains shapes mostly associated with the action start



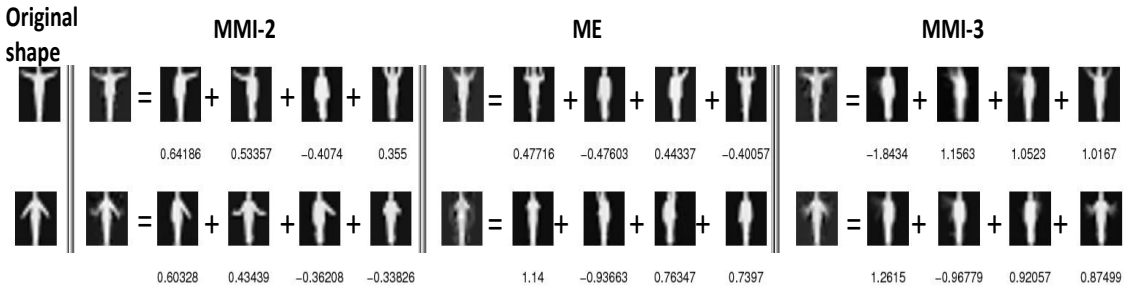
(a) MMI-2 shape attributes



(b) ME shape attributes



(c) MMI-3 shape attributes



(d) Description to two example frames in an unknown action *flap* using attribute dictionaries (Sparsity 10 is used and top-4 attributes are shown.)

Figure 2.3: Learned attribute dictionaries on shape features ("unseen" classes: *flap*, *stop both* and *attention both*)

and end as discussed in Sec. 2.4, which often results in high reconstruction errors shown in Fig. 2.3d. MMI-3 dictionary (Fig. 2.3c) only concerns about the discriminability, thus obvious redundancy can be observed in its dictionary. We can see from Fig. 2.3d, though the action *flap* is unknown to the dictionary, we still obtain a nearly perfect reconstruction through MMI-2, i.e., we can perfectly describe it using attributes in dictionary with corresponding sparse coefficients.

2.6.1.3 Recognition Accuracy

In all of our experiments, we use the following classification schemes: when the global features, i.e., *shape* and *motion*, are used for attribute dictionaries, we first adopt dynamic time warping (DTW) to align and measure the distance between two action sequences in the sparse code domain; then a k-NN classifier is used for recognition. When the local feature *STIP* [2] is used, DTW becomes not applicable, and we simply perform recognition using a k-NN classifier based on the sparse code histogram of each action sequence.

In Fig. 2.4, we present the recognition accuracy on the Keck gesture dataset with different dictionaries sizes and over different global and local features. We use a leave-one-person-out setup, i.e., sequences performed by a person are left out, and report the average accuracy. We choose an initial dictionary size $|D^o|$ to be twice the dimension of an input signal and sparsity 10 is used in this set of experiments. In all cases, the proposed MMI-2 outperforms the rest. The sparse code noise has more effects on the DTW methods than the histogram method, thus, MMI-2 brings more improvements on global features over local features. The peak recognition accuracy obtained from MMI-2 is comparable to 92.86% (motion), 92.86% (shape), 95.24% (shape and motion) reported in [39].

As discussed, the near-optimality of our approach can be guaranteed if the initial dictionary size $|D^o|$ is sufficiently larger than $2|D^*|$. We usually choose a size for D^* to keep $|D^o|$ be 10 to 20 times larger. As shown in Fig. 2.4, such dictionary size range usually produces good recognition performance. We can also decide $|D^*|$ when the MI increase

in (2.8) is below a predefined threshold, which can be obtained via cross validation from training data.

2.6.2 Discriminability of Learned Action Attributes

In this section, we further evaluate the discriminative power of learned action attributes using MMI-2.

2.6.2.1 Recognizing Unknown Actions

The Weizmann human action dataset contains 10 different actions: bend, jack, jump, pjump, run, side, skip, walk, wave1, wave2. Each action is performed by 9 different people. We use the shape and the motion features for attribute dictionaries. In the experiments on the Weizmann dataset, we learn a 50 size dictionary from a 1000 size initial dictionary and the sparsity 10 is used. When we use a leave-one-person-out setup, we obtain 100% recognition accuracy for the Weizmann dataset.

To evaluate the recognition performance of attribute representation for unknown actions, we use a leave-one-action-out setup for dictionary learning, and then use a leave-one-person-out setup for recognition. In this way, one action class is kept unknown to the learned attribute dictionary, and its sparse representation using attributes learned from the rest classes is used for recognition. The recognition accuracy is shown in Table 2.1.

It is interesting to notice from the second row of Table 2.1 that only *jump* can not be perfectly described using attributes learned from the rest 9 actions, i.e., *jump* is described by a set of attributes not completely provided by the rest actions. By examining the

dataset, it is easy to notice *jump* does exhibit unique shapes and motion patterns.

As we see from the third row of the table, omitting attributes of the *wave2*, i.e., the *wave-two-hands* action, brings down the overall accuracy most. Further investigation tells us, when the *wave2* attributes are not present, such accuracy loss is caused by 33% *pjump* being misclassified as *jack*, which means the attributes contributed by *wave2* are useful to distinguish *pjump* from *jack*. This makes great sense as *jack* is very similar to *pjump* but *jack* contains additional *wave-two-hands* pattern.

Unknown Action	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
Action Accuracy	1.00	1.00	0.89	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Overall Accuracy	1.00	1.00	0.98	0.98	1.00	1.00	1.00	0.99	0.97	0.94

Table 2.1: Recognition accuracy on the Weizmann dataset using a leave-one-action-out setup for dictionary learning. The second row is the recognition accuracy on the unknown action, and the third row is the overall average accuracy over all classes given the unknown action. The second row reflects the importance of attributes learned from the rest actions to represent the unknown action, and the third row reflects the importance of attributes from the unknown action to represent the rest actions.

2.6.2.2 Recognizing Realistic Actions

The UCF sports dataset is a set of 150 broadcast sports videos and contains 10 different actions shown in Fig. 2.5. It is a challenging dataset with significant variations in scene content and viewpoints. As the UCF dataset often involves multiple people in the scene,

we use tracks from ground-truth annotations. We use the HOG and the motion features for attribute dictionaries. We learned a 60 size dictionary from a 1200 size initial dictionary and the sparsity 10 is used. We adopt a five-fold cross-validation setup. With such basic features and a simple k-NN classifier, we obtain 83.6% average recognition accuracy over the UCF sports action dataset, and the confusion matrix is shown in Fig. 2.6.

2.6.3 Attribute dictionary on high-level features

We learn our sparse attribute dictionary from features. As discussed in Sec. 2.2, human actions are typically represented by low- or mid-level features, which contain little semantic meanings. Recent advances in action representations suggest the inclusion of semantic information for high-level action features. A promising high-level action feature, ActionBank, is introduced in [42]. The ActionBank representation is a concatenation of max-pooled detection features from many individual action detectors sampled broadly in a semantic space. As reported in [42], the action recognition accuracy using ActionBank features is better than the state of the art, better by 3.7% on UCF Sports, and 10% on UCF50.

In this section, we demonstrate that our learned action attributes can not only benefit from but also enhance high-level features in terms of discriminability. We perform experiments on the UCF Sports and UCF50 action datasets.

We revisit the UCF sports dataset. Instead of the low-level HOG and motion features, we adopt the ActionBank high-level features for attribute dictionaries. A 29930 dimensional ActionBank feature is extracted for each action, and such feature is reduced

to 128 dimensions through PCA. Then, we learned a 40-sized attibute dictionary from a 128-sized initial dictionary and the sparsity 20 is used. We use the same leave-one-out cross-validation setup as [42] for action recognition. In order to emphasize the discriminability of learned action attributes, we adopt a simple k-NN classifier.

The recognition accuracies using high-level ActionBank features are reported in the second part of Table 2.2. We obtain 90.7% by using ActionBank features directly with a k-NN classifier. The recognition accuracy using the initial K-SVD dictionary on ActionBank features is 52.1%. The recognition accuracy using the attribute dictionaries learned by MMI-1, MMI-2 and MMI-3 are 93.6%, 91.5% and 87.9%. We made the following three observations: first, the proposed dictionary learning method significantly enhances dictionary discriminability (better by 41.5% than the initial K-SVD dictionary). Second, the learned attributes using MMI-1 further improve the state of the art discriminability of ActionBank features (better by 3.0%). Third, discriminability improvements from considering class distribution during dictionary learning are less significant while using high-level features, comparing to low-level ones. This can be due to that high-level features like ActionBank have already encoded such semantic information, i.e., the feature appearance carries class information. Though MMI-2 significantly outperforms both MMI-2 and MMI-3 given low-level features, MMI-1 is preferred when high-level semantic features are used.

We conduct another set of experiments using high-level features on the UCF50 action dataset. UCF50 is a very challenging action dataset with 50 action categories, consisting of 6617 realistic videos taken from youtube. Sample frames from the UCF50 action dataset are shown in Fig. 2.7. A 14965 dimensional ActionBank feature is first

Method	Accuracy (%)
Rodriguez et al. [46]	69.2
Yeffet and Wolf [48]	79.3
MMI-2 (HOG&motion)	83.6
Varma and Babu [49]	85.2
Wang et al. [50]	85.6
Le et al. [51]	86.5
Kovashka and Grauman [52]	87.3
Wu et al. [53]	91.3
K-SVD	52.1
MMI-3	87.9
ActionBank	90.7
MMI-2	91.5
MMI-1	**93.7**

Table 2.2: Recognition accuracies on the UCF Sports dataset using high-level features.

extracted for each action, and such feature is reduced to 512 dimensions through PCA. Then, we learned a 128-sized dictionary from a 2048-sized initial dictionary and the sparsity 60 is used. We use 5-fold group-wise cross-validation setup suggested in [42] for action recognition. Again, we adopt a simple k-NN classifier. We obtain 36.7% by using ActionBank features directly with a k-NN classifier, and 41.5% by using the MMI-1 attribute dictionaries learned from ActionBank features. The learned action attributes further improve the discriminability of ActionBank features by 4.8%.

2.6.4 Action Sampling/Summarization using MMI-1

This section presents experiments demonstrating action summarization using the proposed MMI-1 algorithm. We first use the MPEG shape dataset [54] to provide an objective assessment of diversity and coverage enforced by the MMI-1 sampling scheme. Then we provide action summarization examples using the UCF sports dataset.

As discussed in Sec 2.2, actions are described using features extracted from an action interest region. Global action features are typically shape-based or motion-based descriptors. As video summarization often lacks of objective assessment schemes, shape sampling provides an objective alternate to measure diversity and coverage of a sampling/summarization method.

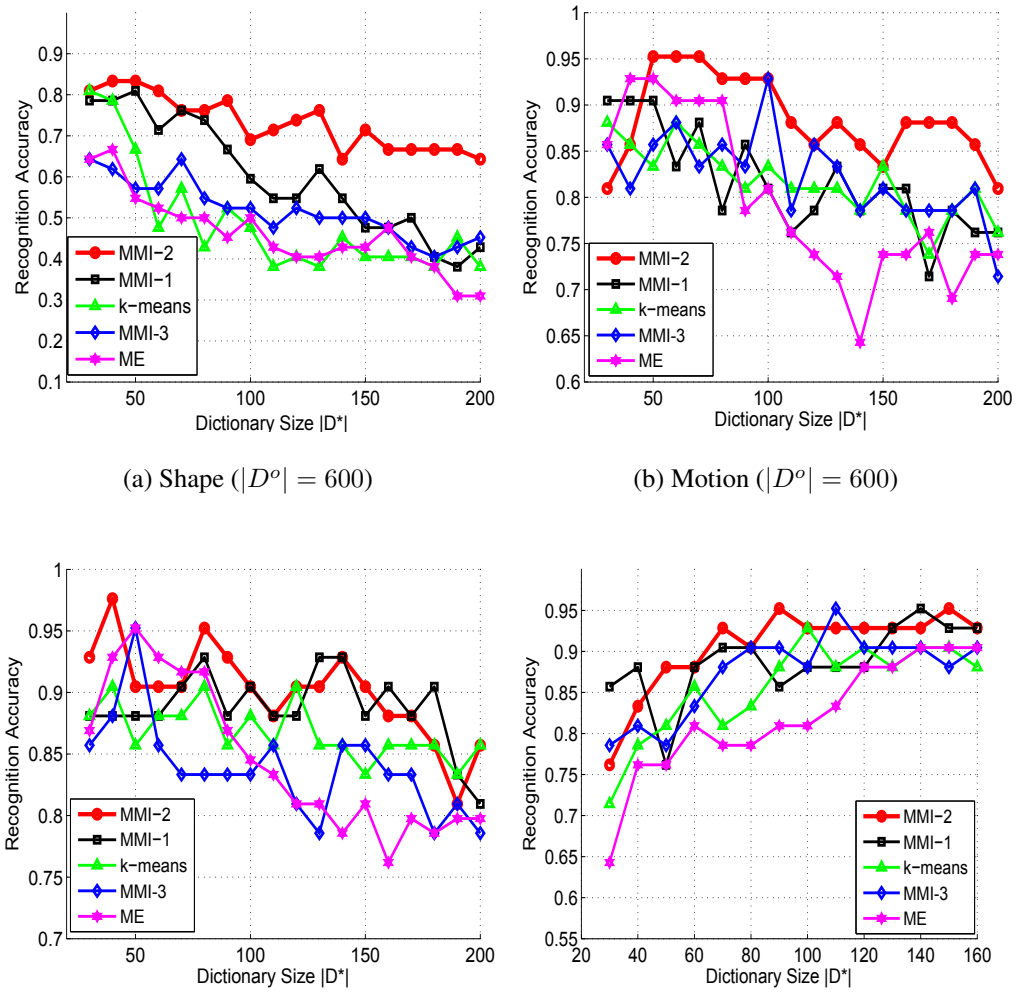
We conducted shape sampling experiments on the MPEG dataset. This dataset contains 70 shape classes with 20 shapes each. As shown in Fig. 2.8a, we use 10 classes with 10 shape each in our experiments. To emphasize both diversity and coverage criteria, we keep our shape descriptor be variant to affine transformations. Thus, shapes with distinct rotation, scaling or translation are considered as outliers. The Top-10 shape sampling results using ME in Fig. 2.8b, which only considers diversity, retrieved 3 classes. The sampling results using k-means in Fig. 2.8c, which focuses on coverage, retrieved 7 classes. As shown in Fig. 2.8d, the sampling results using the proposed MMI-1 method, which enforces both diversity and coverage criteria, retrieved all 10 classes.

In Fig. 2.9, we provide an action summarization example using the proposed MMI-1 method. For the dive sequence in Fig. 2.9a, we describe each frame of the action using both the HOG and the motion features. Then we sample Top-10 frames using MMI-1 and

sort them by timestamps, as shown in Fig. 2.9b. Through a subjective assessment, the dive action summarized using MMI-1 in Fig. 2.9b is compact yet representative.

2.7 Conclusion

We presented an attribute dictionary learning approach via information maximization for action recognition and summarization. By formulating the mutual information for appearance information and class distributions between the learned dictionary and the rest of dictionary space into an objective function, we can ensure the learned dictionary is both representative and discriminative. The objective function is optimized through a GP model proposed for sparse representation. The sparse representation for signals enable the use of kernels locality in GP to speed up the optimization process. An action sequence is described through a set of action attributes, which enable both modeling and recognizing actions, even including "unseen" human actions. Our future work includes how to automatically update the learned dictionary for a new action category.



(a) Shape ($|D^o| = 600$)

(b) Motion ($|D^o| = 600$)

(c) Shape and Motion ($|D^o| = 1200$)

(d) STIP ($|D^o| = 600$)

Figure 2.4: Recognition accuracy on the Keck gesture dataset with different features and dictionary sizes (*shape* and *motion* are global features. *STIP* [2] is a local feature.). The recognition accuracy using initial dictionary D^o: (a) 0.23 (b) 0.42 (c) 0.71 (d) 0.81. In all cases, the proposed MMI-2 (red line) outperforms the rest.

Figure 2.5: Sample frames from the UCF sports action dataset. The actions include: diving, golfing, kicking, weight-lifting, horse-riding, running, skateboarding, swinging-1 (on the pommel horse and on the floor), swinging-2 (at the high bar), walking.

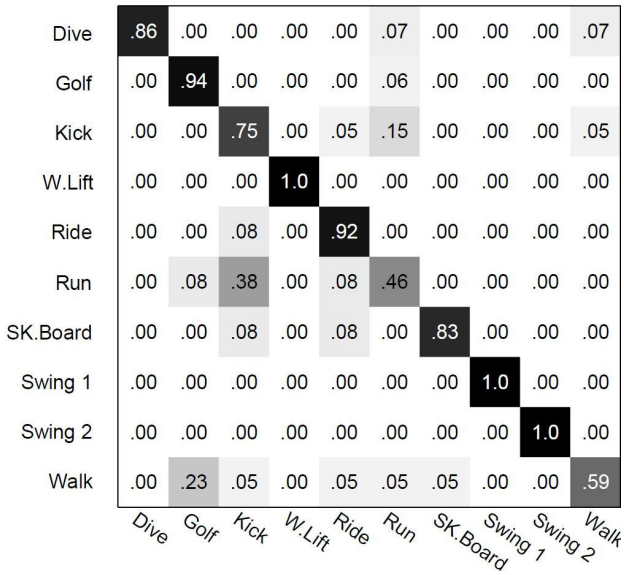


Figure 2.6: Confusion matrix for UCF sports dataset

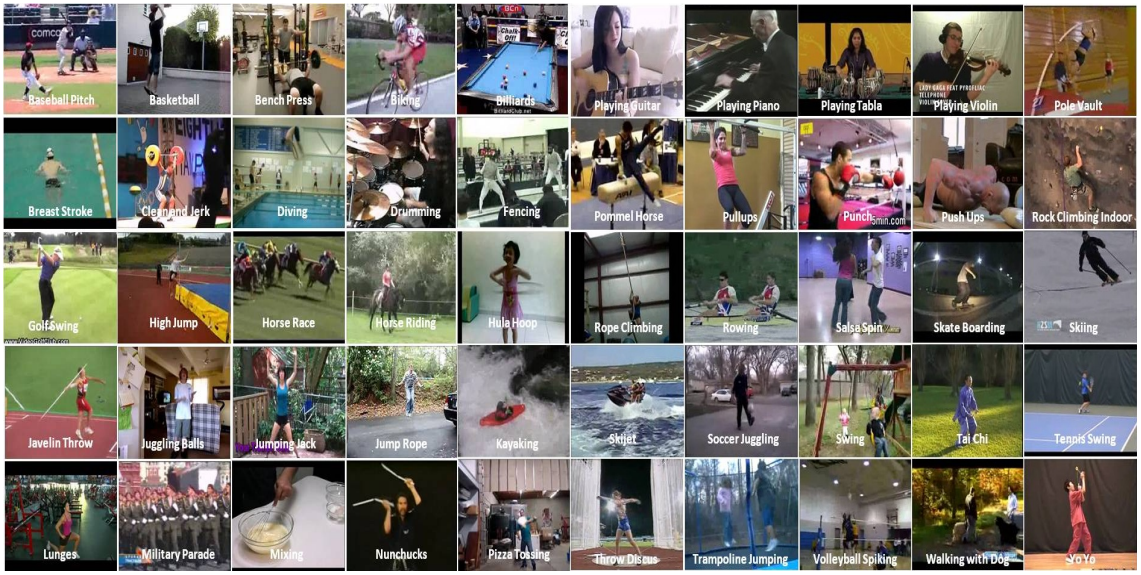
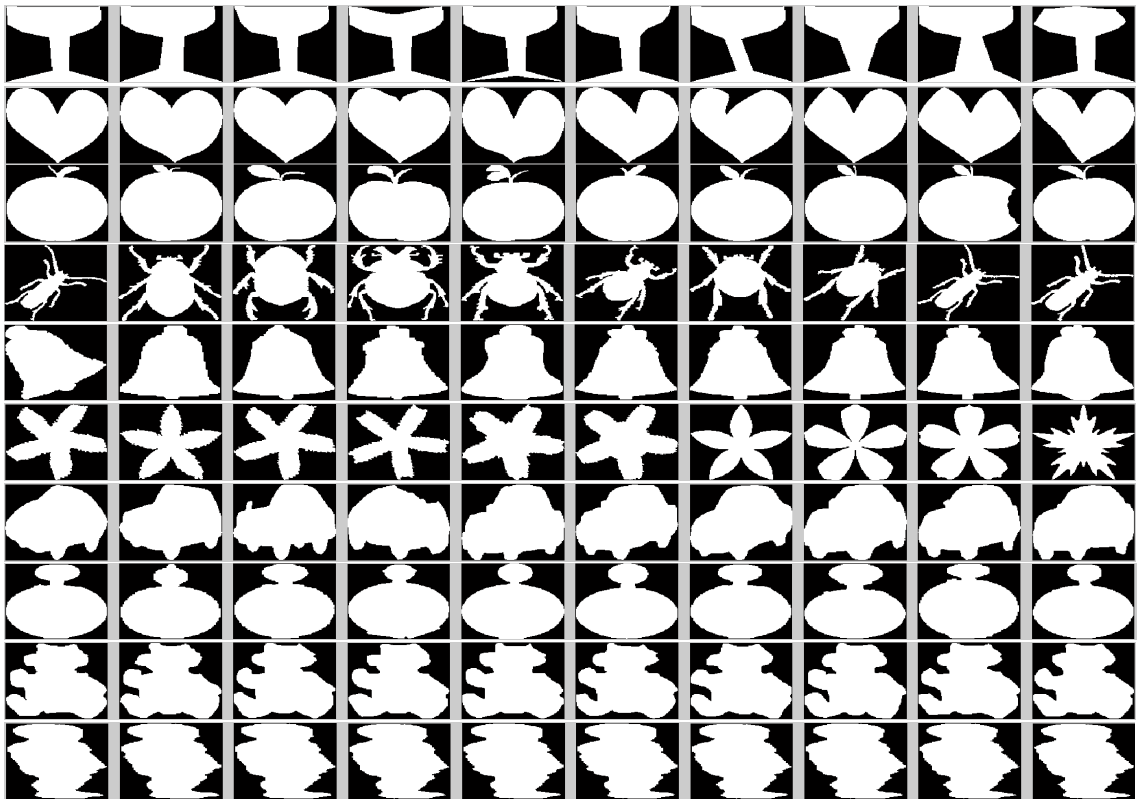


Figure 2.7: Sample frames from the UCF50 action dataset. UCF50 is an action recognition dataset with 50 action categories, consisting of 6617 realistic videos taken from youtube.



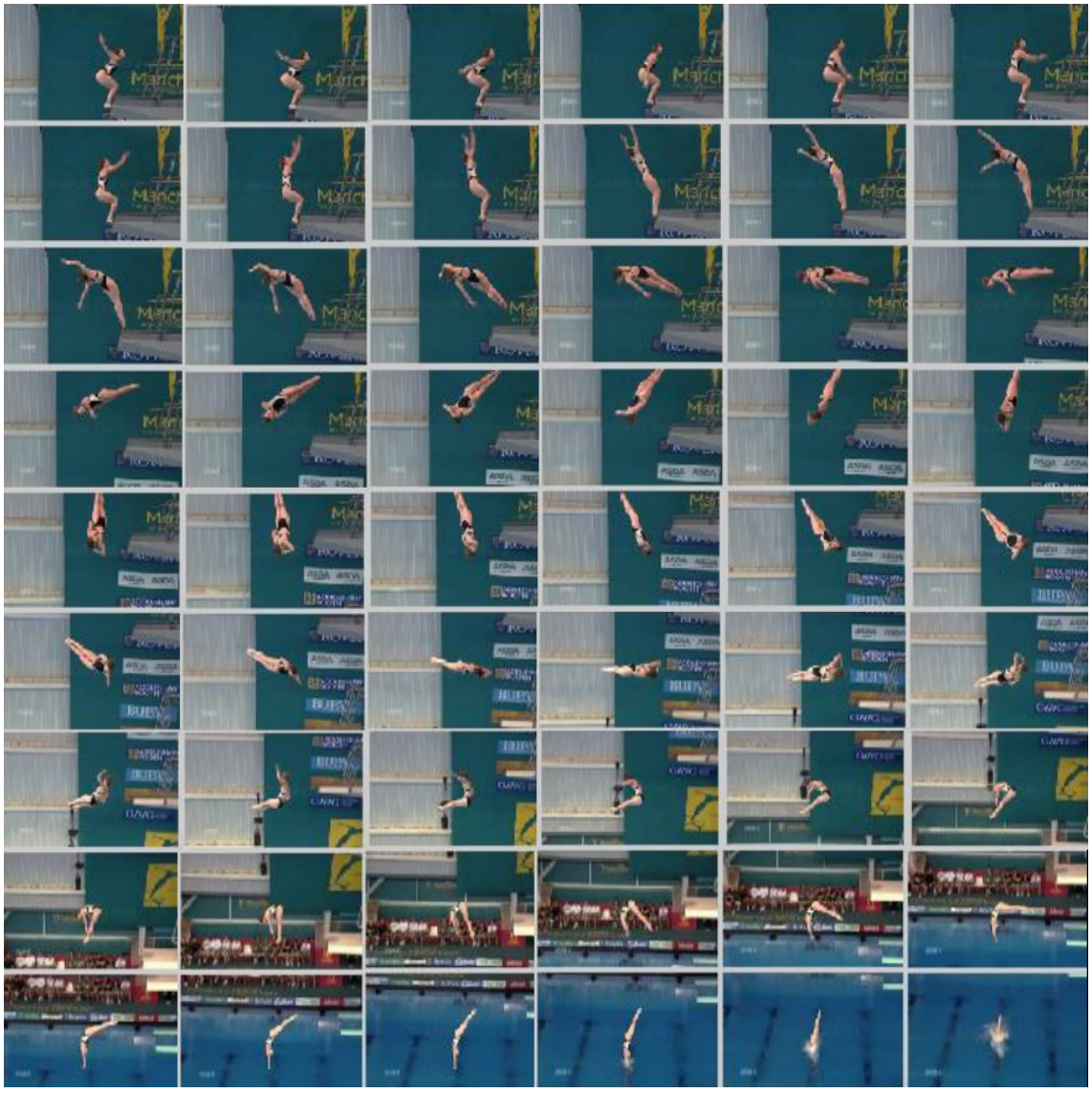
(a) 10 classes from MPEG shape dataset

(b) Top-10 shapes sampled using ME

(c) Top-10 shapes sampled using k-means

(d) Top-10 shapes sampled using the proposed MMI-1

Figure 2.8: Shape sampling on the MPEG dataset. The proposed MMI-1 method, which enforces both diversity and coverage criteria, retrieved all 10 shape classes.



(a) A UCF sports sample dive sequence

(b) A dive action summary obtained using MMI-1

Figure 2.9: An MMI-1 action summarization example using the UCF sports dataset

Chapter 3

Information-theoretic Dictionary Learning

3.1 Introduction

Sparse signal representations have recently drawn much traction in vision, signal and image processing [55], [56], [57], [58]. This is mainly due to the fact that signals and images of interest can be sparse in some dictionary. Given a redundant dictionary $\mathbf{D}$ and a signal $\mathbf{y}$, finding a sparse representation of $\mathbf{y}$ in $\mathbf{D}$ entails solving the following optimization problem

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ subject to } \mathbf{y} = \mathbf{D}\mathbf{x}, \tag{3.1}$$

where the ℓ_0 sparsity measure $\|\mathbf{x}\|_0$ counts the number of nonzero elements in the vector $\mathbf{x}$. Problem (3.1) is NP-hard and cannot be solved in a polynomial time. Hence, approximate solutions are usually sought [57], [59], [60], [61].

The dictionary $\mathbf{D}$ can be either based on a mathematical model of the data [57] or it can be trained directly from the data [62]. It has been observed that learning a dictionary directly from training rather than using a predetermined dictionary (such as wavelet or Gabor) usually leads to better representation and hence can provide improved results in many practical applications such as restoration and classification [55], [56], [58], [63].

Various algorithms have been developed for the task of training a dictionary from examples. One of the most commonly used algorithms is the K-SVD algorithm [20].

Given a set of examples $\{\mathbf{y}_i\}_{i=1}^n$, K-SVD finds a dictionary $\mathbf{D}$ that provides the best representation for each example in this set by solving the following optimization problem

$$(\hat{\mathbf{D}}, \hat{\mathbf{X}}) = \arg\min_{\mathbf{D},\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \text{ subject to } \forall i \ \|\mathbf{x}_i\|_0 \leq T_0, \tag{3.2}$$

where $\mathbf{x}_i$ represents the i^{th} column of $\mathbf{X}$, $\mathbf{Y}$ is the matrix whose columns are $\mathbf{y}_i$ and T_0 is the sparsity parameter. Here, the Frobenius norm is defined as $\|\mathbf{A}\|_F = \sqrt{\sum_{ij} A_{ij}^2}$. The K-SVD algorithm alternates between sparse-coding and dictionary update steps. In the sparse-coding step, $\mathbf{D}$ is fixed and the representation vectors $\mathbf{x}_i$s are found for each example $\mathbf{y}_i$. Then, the dictionary is updated atom-by-atom in an efficient way.

Dictionaries can be trained for both reconstruction and discrimination applications. In the late nineties, Etemand and Chellappa proposed a linear discriminant analysis (LDA) based basis selection and feature extraction algorithm for classification using wavelet packets [64]. Recently, similar algorithms for simultaneous sparse signal representation and discrimination have also been proposed in [37, 38, 65, 66]. Some of the other methods for learning discriminative dictionaries include [4, 24, 26, 65, 67–69]. Additional techniques may be found within these references.

In this chapter, we propose a general method for learning dictionaries for image classification tasks via information maximization. Unlike other previously proposed dictionary learning methods that only consider learning only reconstructive and/or discriminative dictionaries, our algorithm can learn reconstructive, compact and discriminative dictionaries simultaneously. Sparse representation over a dictionary with coherent atoms has the multiple representation problem. A compact dictionary consists of incoherent atoms, and encourages similar signals, which are more likely from the same class, to be

consistently described by a similar set of atoms with similar coefficients [4]. A discriminative dictionary encourages signals from different classes to be described by either a different set of atoms, or the same set of atoms but with different coefficients [37, 38, 68]. Both aspects are critical for classification using sparse representation. The additional reconstructive requirement to a compact and discriminative dictionary enhances the robustness of the discriminant sparse representation [37]. All these three criteria are critical for classification using sparse representation.

Our method of training dictionaries consists of two main stages involving greedy atom selection and simple gradient ascent atom updates, resulting in a highly efficient algorithm. In the first stage, dictionary atoms are selected in a greedy way such that the common internal structure of signals belonging to a certain class is extracted while at the same time ensuring global discrimination among the different classes. In the second stage, the dictionary is updated for improved discrimination and reconstruction via a simple gradient ascent method that maximizes the mutual information (MI) between the signals and the dictionary, as well as the sparse coefficients and the class labels.

Fig. 3.1 presents a comparison in terms of the discriminative power of the information-theoretic dictionary learning approach presented in this chapter with three state-of-the-art methods. Scatter plots of sparse coefficients obtained using the different methods show that our method provides more discriminative sparse representation, leading to significantly better classification accuracy.

This chapter makes the following contributions:

- We propose a two-stage information-theoretic dictionary learning framework for

image classification tasks.

- We learn reconstructive, compact and discriminative dictionaries simultaneously.
- We achieve an efficient dictionary learning algorithm through greedy atom selection and simple gradient ascent atom updates.

The organization of the chapter is as follows. Section 3.2 defines and formulates the information theoretic dictionary learning problem. In Section 3.3, the proposed dictionary learning algorithm is detailed. Experimental results are presented in Section 3.4 and Section 3.5 concludes the chapter with a brief summary and discussion.

3.2 Background and Problem Formulation

Suppose we are given a set of N signals (images) in an n-dim feature space $\mathbf{Y} = [\mathbf{y_1}, ..., \mathbf{y_N}],\ \mathbf{y_i} \in \mathbb{R}^n$. Given that signals are from p distinct classes and N_c signals are from the c-th class, $c \in \{1, \cdots, p\}$, we denote $\mathbf{Y} = \{\mathbf{Y}_c\}_{c=1}^p$, where $\mathbf{Y}_c = [\mathbf{y}_1^c, \cdots, \mathbf{y}_{N_c}^c]$ are signals in the c-th class. When the class information is relevant, similarly, we define $\mathbf{X} = \{\mathbf{X}_c\}_{c=1}^p$, where $\mathbf{X}_c = [\mathbf{x}_1^c, \cdots, \mathbf{x}_{N_c}^c]$ is the sparse representation of $\mathbf{Y}_c$.

Given a sample $\mathbf{y}$ at random, the entropy (uncertainty) of the class label in terms of class prior probabilities is defined as

$$H(C) = \sum_c p(c) \left(\frac{1}{p(c)} \right).$$

The mutual information which indicates the decrease in uncertainty about the pattern $\mathbf{y}$ due to the knowledge of the underlying class label c is defined as

$$I(\mathbf{Y}; C) = H(\mathbf{Y}) - H(\mathbf{Y}|C),$$

where $H(\mathbf{Y}|C)$ is the conditional entropy defined as

$$H(\mathbf{Y}|C) = \sum_{\mathbf{y},c} p(\mathbf{y}, c) \log \left(\frac{1}{p(\mathbf{y}|c)} \right).$$

Given $\mathbf{Y}$ and an initial dictionary $\mathbf{D}^o$ with ℓ_2 normalized columns, we aim to learn a compact, reconstructive and discriminative dictionary $\mathbf{D}^*$ via maximizing the mutual information between $\mathbf{D}^*$ and the unselected atoms $\mathbf{D}^o \backslash \mathbf{D}^*$ in $\mathbf{D}^o$, between the sparse codes $\mathbf{X}_{\mathbf{D}^*}$ associated with $\mathbf{D}^*$ and the signal class labels C, and finally between the signals $\mathbf{Y}$ and $\mathbf{D}^*$, i.e.,

$$\arg\max_{\mathbf{D}} \lambda_1 I(\mathbf{D}; \mathbf{D}^o \backslash \mathbf{D}) + \lambda_2 I(\mathbf{X}_{\mathbf{D}}; C) + \lambda_3 I(\mathbf{Y}; \mathbf{D}) \tag{3.3}$$

where $\{\lambda_1, \lambda_2, \lambda_3\}$ are the parameters to balance the contributions from compactness, discriminability and reconstruction terms, respectively.

It is widely known that inclusion of additional criteria, such as a discriminative term, in a dictionary learning framework often involves challenging optimization algorithms [65, 68, 69]. As discussed above, compactness, discriminability and reconstruction terms are all critical for classification using sparse representation. Maximizing mutual information enables a simple way to unify all three criteria for dictionary learning. As suggested in [43] and [4], maximizing mutual information can also lead to a sub-modular objective function, i.e., a greedy yet near-optimal approach, for dictionary learning.

A two-stage approach is adopted to satisfy (3.3). In the first stage, each term in (3.3) is maximized in a unified greedy manner and involves a closed-form evaluation, thus atoms can be greedily selected from the initial dictionary while satisfying (3.3). In the second stage, the selected dictionary atoms are updated using a simple gradient ascent

method to further maximize

$$\lambda_2 I(\mathbf{X}_{\mathbf{D}}; C) + \lambda_3 I(\mathbf{Y}; \mathbf{D}).$$

3.3 Information-theoretic Dictionary Learning

In this section, we present the details of our *Information-theoretic Dictionary Learning* (**ITDL**) approach for classification tasks. The dictionary learning procedure is divided into two main steps: *Information-theoretic Dictionary Selection* (**ITDS**) and *Information-theoretic Dictionary Update* (**ITDU**). In what follows, we describe these steps in detail.

3.3.1 Dictionary Selection

Given input signals $\mathbf{Y}$ and an initial dictionary $\mathbf{D}^o$, we select a subset of dictionary atoms $\mathbf{D}^*$ from $\mathbf{D}^o$ via information maximization, i.e., maximizing (3.3), to encourage the signals from the same class to have very similar sparse representation yet have the discriminative power. In this section, we illustrate why each term in (3.3) describes the dictionary compactness, discrimination and representation, respectively. We also show that how each term in (3.3) can be maximized in a unified greedy manner that involves closed-form computations. Therefore, if we start with $\mathbf{D}^* = \emptyset$, and greedily select the next best atom $\mathbf{d}^*$ from $\mathbf{D}^o \backslash \mathbf{D}^*$ which provides an information increase to (3.3), we obtain a set of dictionary atoms that is compact, reconstructive and discriminative at the same time. To this end, we consider in detail each term in (3.3) separately.

3.3.1.1 Dictionary compactness $I(\mathbf{D}^*; \mathbf{D}^o \backslash \mathbf{D}^*)$

The dictionary compactness $I(\mathbf{D}^*; \mathbf{D}^o \backslash \mathbf{D}^*)$ has been studied in our early work [4]. We summarize [4] to complete our information-driven dictionary selection discussion. [4] suggests dictionary compactness is required to avoid the multiple sparse representation problem for better classification performance. In [4], we first model sparse representation through a Gaussian Process model to define the mutual information $I(\mathbf{D}^*; \mathbf{D}^o \backslash \mathbf{D}^*)$. A compact dictionary can be then obtained as follows: we start with $\mathbf{D}^* = \emptyset$ and iteratively choose the next best dictionary item $\mathbf{d}^*$ from $\mathbf{D}^o \backslash \mathbf{D}^*$ which provides a maximum increase in mutual information, i.e.,

$$\arg\max_{\mathbf{d}^* \in \mathbf{D}^o \backslash \mathbf{D}^*} I(\mathbf{D}^* \cup \mathbf{d}^*; \mathbf{D}^o \backslash (\mathbf{D}^* \cup \mathbf{d}^*)) - I(\mathbf{D}^*; \mathbf{D}^o \backslash \mathbf{D}^*). \tag{3.4}$$

It has been proved in [43] that the above greedy algorithm serves a polynomial-time approximation that is within $(1 - 1/e)$ of the optimum.

3.3.1.2 Dictionary Discrimination $I(\mathbf{X}_{\mathbf{D}^*}; C)$

Using any pursuit algorithm such as OMP [60], we initialize the sparse coefficients $\mathbf{X}_{\mathbf{D}^o}$ for input signals $\mathbf{Y}$ and an initial dictionary $\mathbf{D}^o$. Given $\mathbf{X}_{\mathbf{D}^*}$ are sparse coefficients associated with the desired set of atoms $\mathbf{D}^*$ and C are the class labels for input signals $\mathbf{Y}$, based on [70], an upper bound on the Bayes error over sparse representation $E(\mathbf{X}_{\mathbf{D}^*})$ is obtained as

$$\frac{1}{2}(H(C) - I(\mathbf{X}_{\mathbf{D}^*}; C)).$$

This bound is minimized when $I(\mathbf{X}_{\mathbf{D}^*}; C)$ is maximized. Thus, a discriminative dictionary $\mathbf{D}^*$ is obtained via

$$\arg\max_{\mathbf{D}^*} I(\mathbf{X}_{\mathbf{D}^*}; C). \tag{3.5}$$

We maximize (3.5) using a greedy algorithm initialized by $\mathbf{D}^* = \emptyset$ and iteratively choosing the next best dictionary atom $\mathbf{d}^*$ from $\mathbf{D}^o \backslash \mathbf{D}^*$ which provides a maximum mutual information increase, i.e.,

$$\arg \max_{\mathbf{d}^* \in \mathbf{D}^o \backslash \mathbf{D}^*} I(\mathbf{X}_{\mathbf{D}^* \cup \mathbf{d}^*}; C) - I(\mathbf{X}_{\mathbf{D}^*}; C), \tag{3.6}$$

where $I(\mathbf{X}_{\mathbf{D}^*}; C)$ is evaluated as follows

$$\begin{aligned} I(\mathbf{X}_{\mathbf{D}^*}; C) &= H(\mathbf{X}_{\mathbf{D}^*}) - H(\mathbf{X}_{\mathbf{D}^*}|C) \\ &= H(\mathbf{X}_{\mathbf{D}^*}) - \sum_{c=1}^{p} p(c) H(\mathbf{X}_{\mathbf{D}^*}|c). \end{aligned} \tag{3.7}$$

Entropy measures in (3.7) involve computation of probability density functions $p(\mathbf{X}_{\mathbf{D}^*})$ and $p(\mathbf{X}_{\mathbf{D}^*}|c)$. We adopt the kernel density estimation method [71] to non-parametrically estimate the probability densities. Using isotropic Gaussian kernels (i.e. $\mathbf{\Sigma} = \sigma^2 \mathbf{I}$, where $\mathbf{I}$ is the identity matrix), the class dependent density for the c-th class can be estimated as

$$p(\mathbf{x}|c) = \frac{1}{N_c} \sum_{j=1}^{N_c} \mathbb{K}_G(\mathbf{x} - \mathbf{x}_j^c, \sigma^2 \mathbf{I}), \tag{3.8}$$

where $\mathbb{K}_G$ is a d-dim Gaussian kernel defined as

$$\mathbb{K}_G(\mathbf{x}, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \mathbf{x}^T \mathbf{\Sigma}^{-1} \mathbf{x}\right). \tag{3.9}$$

With $p(c) = \frac{N_c}{N}$, we can estimate $p(\mathbf{x})$ as

$$p(\mathbf{x}) = \sum_c p(\mathbf{x}|c) p(c).$$

3.3.1.3 Dictionary Representation $I(\mathbf{Y}; \mathbf{D}^*)$

A representative dictionary $\mathbf{D}^*$ maximizes the mutual information between dictionary atoms and the signals, i.e.,

$$\arg\max_{\mathbf{D}^*} I(\mathbf{Y}; \mathbf{D}^*). \tag{3.10}$$

We obtain a representative dictionary via a similar greedy manner as discussed above. That is, we iteratively choose the next best dictionary atom $\mathbf{d}^*$ from $\mathbf{D}^o \backslash \mathbf{D}^*$ which provides the maximum increase in mutual information,

$$\arg\max_{\mathbf{d}^* \in \mathbf{D}^o \backslash \mathbf{D}^*} I(\mathbf{Y}; \mathbf{D}^* \cup \mathbf{d}^*) - I(\mathbf{Y}; \mathbf{D}^*). \tag{3.11}$$

By assuming the signals are drawn independently and using the chain-rule of entropies, we can evaluate $I(\mathbf{Y}; \mathbf{D}^*)$ as

$$\begin{aligned} I(\mathbf{Y}; \mathbf{D}^*) &= H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{D}^*) \\ &= H(\mathbf{Y}) - \sum_{i=1}^{N} H(\mathbf{y}_i|\mathbf{D}^*). \end{aligned} \tag{3.12}$$

$H(\mathbf{Y})$ is independent of dictionary selection and can be ignored. To evaluate $H(\mathbf{y}_i|\mathbf{D}^*)$ in (3.12), we define $p(\mathbf{y}_i|\mathbf{D}^*)$ through the following relation holding for each input signal $\mathbf{y}_i$,

$$\mathbf{y}_i = \mathbf{D}^* \mathbf{x}_i + \mathbf{r}_i,$$

where $\mathbf{r}_i$ is a Gaussian residual vector with variance σ_r^2. Such a relation can be written in a probabilistic form as,

$$p(\mathbf{y}_i|\mathbf{D}^*) \propto \exp(-\frac{1}{2\sigma_r^2}||\mathbf{y}_i - \mathbf{D}^* \mathbf{x}_i||^2).$$

3.3.1.4 Selection of λ_1, λ_2 and λ_3

The parameters λ_1, λ_2 and λ_3 in (3.3) are data dependent and can be estimated as the ratio between the maximal information gained from an atom to the respective compactness, discrimination and reconstruction measure, i.e.,

$$\begin{aligned} \lambda_1 &= 1, \\ \lambda_2 &= \frac{\max_i I(\mathbf{X}_{\mathbf{d_i}}; C)}{\max_i I(\mathbf{d_i}; \mathbf{D}^o \backslash \mathbf{d_i})}, \\ \lambda_3 &= \frac{\max_i I(\mathbf{Y}; \mathbf{d_i})}{\max_i I(\mathbf{d_i}; \mathbf{D}^o \backslash \mathbf{d_i})}. \end{aligned} \tag{3.13}$$

For each term in (3.3), only the first greedily selected atom based on (3.4), (3.6) and (3.11), respectively are involved in parameter estimation. This leads to an efficient process in finding parameters.

3.3.2 Dictionary Update

A representative and discriminative dictionary $\mathbf{D}$ produces the maximal MI between the sparse coefficients and the class labels, as well as the signals and the dictionary, i.e.,

$$\max_{\mathbf{D}} \lambda_2 I(\mathbf{X}_{\mathbf{D}}; C) + \lambda_3 I(\mathbf{Y}; \mathbf{D}).$$

In the dictionary update stage, we update the set of selected dictionary atoms $\mathbf{D}$ to further enhance the discriminability and representation.

To achieve sparsity, we assume the cardinality of the set of selected atoms $\mathbf{D}$ is much smaller than the dimension of the signal feature space. Under such an assumption, the sparse representation of signals $\mathbf{Y}$ can be obtained as $\mathbf{X}_{\mathbf{D}} = \mathbf{D}^{\dagger}\mathbf{Y}$ which minimizes

the representation error $\|\mathbf{Y} - \mathbf{D}\mathbf{X}_\mathbf{D}\|_F^2$, where

$$\mathbf{D}^\dagger = (\mathbf{D}^\mathbf{T}\mathbf{D})^{-1}\mathbf{D}^\mathbf{T}.$$

Thus, updating dictionary atoms for improving discriminability while maintaining representation is transformed into finding $\mathbf{D}^\dagger$ that maximizes

$$I(\mathbf{D}^\dagger\mathbf{Y}; C).$$

3.3.2.1 A Differentiable Objective Function

To enable a simple gradient ascent method for dictionary update, we first approximate $I(\mathbf{D}^\dagger\mathbf{Y}; C)$ using a differentiable objective function. $I(\mathbf{X}; C)$ can be viewed as the Kullback-Leibler (KL) divergence $D(p\|q)$ between $p(\mathbf{X}, C)$ and $p(\mathbf{X})p(C)$, where $\mathbf{X} = \mathbf{D}^\dagger\mathbf{Y}$. Motivated by [72], we approximate the KL divergence $D(p\|q)$ with the quadratic divergence (QD), defined as

$$Q(p\|q) = \int_t (p(t) - q(t))^2 \, dt,$$

making $I(\mathbf{X}; C)$ differentiable. Due to the property that

$$D(p\|q) \geq \frac{1}{2} Q(p\|q),$$

by maximizing the QD, one can also maximize a lower bound to the KL divergence. With QD, $I(\mathbf{X}; C)$ can now be evaluated as,

$$\begin{aligned} I_Q(\mathbf{X}; C) = &\sum_c \int_\mathbf{x} p(\mathbf{x}, c)^2 \, d\mathbf{x} \\ &- 2\sum_c \int_\mathbf{x} p(\mathbf{x}, c)p(\mathbf{x})p(c) \, d\mathbf{x} \\ &+ \sum_c \int_\mathbf{x} p(\mathbf{x})^2 p(c)^2 \, d\mathbf{x}. \end{aligned} \tag{3.14}$$

In order to evaluate the individual terms in (3.14), we need to derive expressions for the kernel density estimates of various density terms appearing in (3.14). Observe that for the two Gaussian kernels in (3.9), the following holds

$$\int_{\mathbf{x}} \mathbb{K}_G(\mathbf{x}-\mathbf{s}_i, \mathbf{\Sigma}_1)\mathbb{K}_G(\mathbf{x}-\mathbf{s}_j, \mathbf{\Sigma}_2)\, d\mathbf{x} = \mathbb{K}_G(\mathbf{s}_i-\mathbf{s}_j, \mathbf{\Sigma}_1+\mathbf{\Sigma}_2). \tag{3.15}$$

Using (3.8), $p(c) = \frac{N_c}{N}$ and $p(\mathbf{x}, c) = p(\mathbf{x}|c)p(c)$, we have

$$p(\mathbf{x}, c) = \frac{1}{N}\sum_{j=1}^{N_c} \mathbb{K}_G(\mathbf{x}-\mathbf{x}_j^c, \sigma^2\mathbf{I}).$$

Similarly, since $p(\mathbf{x}) = \sum_c p(\mathbf{x}, c)$, we have

$$p(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{K}_G(\mathbf{x}-\mathbf{x}_i, \sigma^2\mathbf{I}).$$

Inserting expressions for $p(\mathbf{x}, c)$ and $p(\mathbf{x})$ into (3.14) and using (3.15), we get the following closed form

$$I_Q(\mathbf{X}; C) = \frac{1}{N^2}\sum_{c=1}^{p}\sum_{k=1}^{N_c}\sum_{l=1}^{N_c} \mathbb{K}_G(\mathbf{x}_k^c - \mathbf{x}_l^c, 2\sigma^2\mathbf{I}) \tag{3.16}$$

$$- \frac{2}{N^2}\sum_{c=1}^{p}\frac{N_c}{N}\sum_{j=1}^{N_c}\sum_{k=1}^{N} \mathbb{K}_G(\mathbf{x}_j^c - \mathbf{x}_k, 2\sigma^2\mathbf{I})$$

$$+ \frac{1}{N^2}\left(\sum_{c=1}^{p}\left(\frac{N_c}{N}\right)^2\right)\sum_{k=1}^{N}\sum_{l=1}^{N} \mathbb{K}_G(\mathbf{x}_k - \mathbf{x}_l, 2\sigma^2\mathbf{I}). \tag{3.17}$$

3.3.2.2 Gradient Ascent Update

For simplicity, we define a new matrix Φ as

$$\Phi \triangleq (\mathbf{D}^\dagger)^{\mathbf{T}}.$$

Once we have estimated $I_Q(\mathbf{X}; C)$ as a function of the data set in a differential form, where $\mathbf{X} = \Phi^T\mathbf{Y}$, we can use gradient ascent on $I_Q(\mathbf{X}; C)$ to search for the optimal Φ

maximizing the quadratic mutual information with

$$\Phi_{k+1} = \Phi_k + \nu \frac{\partial I_Q}{\partial \Phi}|_{\Phi=\Phi_k}$$

where $\nu \geq 0$ defining the step size, and

$$\frac{\partial I_Q}{\partial \Phi} = \sum_{c=1}^{p} \sum_{i=1}^{N_c} \frac{\partial I_Q}{\partial \mathbf{x}_i^c} \frac{\partial \mathbf{x}_i^c}{\partial \Phi}.$$

Since $\mathbf{x}_i^c = \Phi^T \mathbf{y}_i^c$, we get

$$\frac{\partial \mathbf{x}_i^c}{\partial \Phi} = (\mathbf{y}_i^c)^T.$$

Note that

$$\frac{\partial}{\partial \mathbf{x}_i} \mathbb{K}_G(\mathbf{x}_i - \mathbf{x}_j, 2\sigma^2 \mathbf{I}) = \mathbb{K}_G(\mathbf{x}_i - \mathbf{x}_j, 2\sigma^2 \mathbf{I}) \frac{(\mathbf{x}_i - \mathbf{x}_j)}{2\sigma^2}.$$

We have

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}_i^c} I_Q = \frac{1}{N^2\sigma^2} \sum_{k=1}^{N_c} \mathbb{K}_G(\mathbf{x}_k^c - \mathbf{x}_i^c, 2\sigma^2 \mathbf{I})(\mathbf{x}_k^c - \mathbf{x}_i^c) \\ -\frac{2}{N^2\sigma^2} \left(\sum_{c=1}^{p} \left(\frac{N_c}{N} \right)^2 \right) \sum_{k=1}^{N} \mathbb{K}_G(\mathbf{x}_k - \mathbf{x}_i^c, 2\sigma^2 \mathbf{I})(\mathbf{x}_k - \mathbf{x}_i^c) \\ +\frac{1}{N^2\sigma^2} \sum_{k=1}^{p} \frac{N_k + N_c}{2N} \sum_{j=1}^{N_k} \mathbb{K}_G(\mathbf{x}_j^k - \mathbf{x}_i^c, 2\sigma^2 \mathbf{I})(\mathbf{x}_j^k - \mathbf{x}_i^c). \end{aligned} \tag{3.18}$$

Once Φ is updated, the dictionary $\mathbf{D}$ can be updated using the relation $\Phi = (\mathbf{D}^\dagger)^\mathbf{T}$. Such dictionary updates guarantee convergence to a local maximum due to the fact that the quadratic divergence is bounded [73].

3.3.3 Dictionary Learning Framework

Given a dictionary $\mathbf{D}^o$, a set of signals $\mathbf{Y}$, the class labels C and a sparsity level T, the supervised sparse coding method given in Algorithm 1 represents these signals at once

as a linear combination of a common subset of T atoms in $\mathbf{D}$, where T is much smaller than the dimension of the signal feature space to achieve sparsity. We obtain a sparse representation as each signal has no more than T coefficients in its decomposition. The advantage of simultaneous sparse decomposition for classification has been discussed in [37]. Such simultaneous decompositions extract the internal structure of given signals and neglects minor intra-class variations. The ITDS stage in Algorithm 1 ensures such common set of atoms are compact, discriminative and reconstructive.

When the internal structures of signals from different classes can not be well represented in a common linear subspace, Algorithm 2 illustrates supervised sparse coding with a dedicated set of atoms per class. It is noted in Algorithm 2 that both the discriminative and reconstructive terms in ITDS are handled on a class by class basis.

A sparse dictionary learning framework, such as K-SVD [20] which learns a dictionary that minimizes the reconstruction error, usually consists of sparse coding and update stages. In K-SVD, at the coding stage, a pursuit algorithm is employed to select a set of atoms for each signal; and at the update stage, the selected atoms are updated through SVD for improved reconstruction. Similarly, in Algorithm 3, at the coding stage, ITDS is employed to select a set of atoms for each class of signals; and at the update stage, the selected atoms are updated through ITDU for improved reconstruction and discrimination. Algorithm 3 is also applicable to the case when sparse coding is achieved using global atoms.

3.4 Experimental Evaluation

This section presents an experimental evaluation on three public datasets: the Extended YaleB face dataset [74], the USPS handwritten digits dataset [75], and the 15-Scenes dataset [76]. The Extended YaleB dataset contains 2414 frontal face images for 38 individuals. This dataset is challenging due to varying illumination conditions and expressions. The USPS dataset consists of 8-bit 16×16 images of "0" through "9" and 1100 examples for each class. The 15-Scenes dataset contains 4485 images falling into 15 scene categories. The 15 categories include images of living rooms, kitchens, streets, industrials, etc.. In all of our experiments, linear SVMs on the sparse coefficients are used for classifiers. First, we thoroughly evaluate the basic behaviors of the proposed dictionary learning method. Then we evaluate the discriminative power of the ITDL dictionary over the full Extended YaleB dataset, the full USPS dataset, and the 15-Scenes dataset.

3.4.1 Evaluation with Illustrative Examples

To enable visualized illustrations, we conduct the first set of experiments on the first four subjects in the Extended YaleB face dataset and the first four digits in the USPS digit dataset. Half of the data are used for training and the rest is used for testing.

3.4.1.1 Comparing Atom Selection Methods

We initialize a 128 sized dictionary using the K-SVD algorithm [20] on the training face images of the first four subjects in the Extended YaleB dataset. A K-SVD dictionary only minimizes the reconstruction error and is not yet optimal for classification tasks.

Though one can also initialize the dictionary directly with training samples or even with random noise, a better initial dictionary generally helps ITDL in terms of classification performance, due to the fact that an ITDL dictionary converges to a local maximum.

In Fig. 3.2, we present the recognition accuracy and the reconstruction error with different sparsity on the first four subjects in the Extended YaleB dataset. The Root Mean Square Error (RMSE) is employed to measure the reconstruction error. To illustrate the impact of the compactness, discrimination and reconstruction terms in (3.3), we keep one term at a time for the three selection approaches, i.e., the compact, the discriminative and the reconstructive method. The compact method is equivalent to MMI-1 [4].

Parameters λ_1, λ_2 and λ_3 in (3.3) are estimated as discussed in Section 3.3.1.4. As the dictionary learning criteria becomes less critical when sparsity increases, i.e., more energies in signals are actually preserved, we focus on curves in Fig. 3.2 when sparsity$<$20. Although sparse coding methods generally perform well for face recognition, it is still easy to notice that the proposed ITDS method using all three terms (red) significantly outperforms those which optimize just one of the three terms, compactness (black), discrimination (blue), and representation (green), in terms of recognition accuracy. For example, the discrimination term alone (blue) leads to a better initial but poor overall recognition performance. The proposed ITDS method also provides moderate reconstruction error.

It is noted that IDS exhibits comparable recognition accuracy to MMI-2 (pink) [4] with global atoms, and significantly outperforms it with class dedicated atoms. The reason is that, instead of explicitly considering the discriminability of dictionary atoms, MMI-2 enforces the diversity of classes associated with atoms. Such class diversity criteria becomes less effective when there are only two classes in the dedicate atom case. In

Fig. 3.2, it is interesting to note that the reconstructive method delivers nearly identical recognition accuracy and RMSE to SOMP [3] with both the shared and dedicated atoms, given the different formulations of two methods. The proposed dictionary selection using all three terms provides a good local optimum to converge at the dictionary update stage.

3.4.1.2 Enhanced Discriminability with Atom Update

We illustrate how the discriminability of dictionary atoms selected by the ITDS method can be further enhanced using the proposed ITDU method. We initialize a 128 sized K-SVD dictionary for the face images and a 64 sized K-SVD dictionary for the the digit images. Sparsity 2 is adopted for visualization, as the non-zero sparse coefficients of each image can now be plotted as a 2-D point. In Fig. 3.3, with a common set of atoms shared over all classes, sparse coefficients of all samples become points in the same 2-D coordinate space. Different classes are represented by different colors. The original images are also shown and placed at the coordinates defined by their non-zero sparse coefficients. The atoms to be updated in Fig. 3.3a and 3.3d are selected using ITDS. We can see from Fig. 3.3 that the proposed ITDU method makes sparse coefficients of different classes more discriminative, leading to significantly improved classification accuracy. Fig. 3.4 shows that the ITDU method also enhances the discriminability of atoms dedicated to each class. It is noted that, though the dictionary update sometimes only converges after a considerable number of iterations, based on our experience, the first 50 to 100 iterations in general bring significant improvement in classification accuracy.

3.4.1.3 Enhanced Reconstruction with Atom Update

From Fig. 3.5e, we notice obvious errors in the reconstructed digits, shown in Fig. 3.5d with atoms selected from the initial K-SVD dictionary using ITDS. After 30 ITDU iterations, Fig. 3.5f shows that all digits are reconstructed correctly with a unified intra-class structure and limited intra-class variation. This leads to a more accurate classification as shown in Fig. 3.4. It is noted that Fig. 3.5 and Fig. 3.4 are results from the same set of experiments. As can be seen from Fig. 3.5g, after ITDU converges, all digits are reconstructed correctly with the true underlying intra-class structures, i.e., the left-slanted and right-slanted styles for both digits "1" and "0". Fig. 3.5h shows the images in Fig. 3.5d with 60% missing pixels. The recognition rate for Fig. 3.5i, Fig. 3.5j, and Fig. 3.5k are 76.87%, 85.03% and 85.71%, respectively.

3.4.2 Discriminability of ITDL Dictionaries

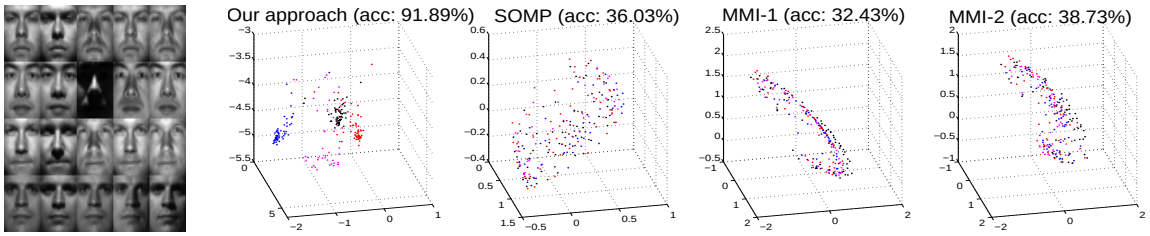
We evaluate the discriminative power of ITDL dictionaries over the complete USPS dataset, where we use 7291 images for training and 2007 images for testing, and the Extended YaleB face dataset, where we randomly select half of the images as training and the other half for testing, and finally the 15-Scenes dataset, where we randomly use 100 images per class for training and used the remaining data for testing.

For each dataset, we initialize a 512 sized dictionary from K-SVD and set the sparsity to be 30. Then we perform 30 iterations of dictionary update and report the peak classification performance. Here we adopt a dedicated set of atoms for each class and input the concatenated sparse representation into a linear SVM classifier. For the Extended

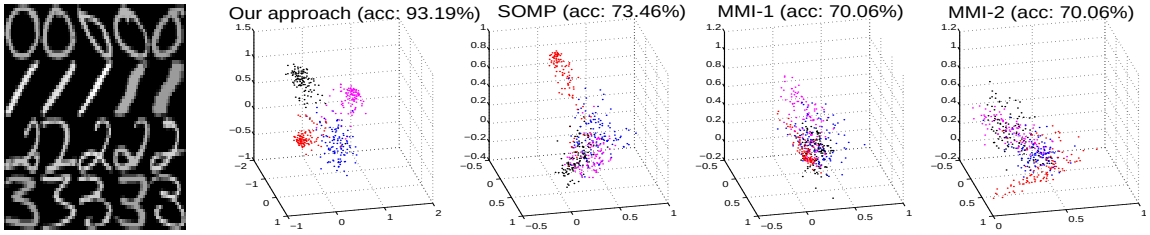
YaleB face dataset, we adopt the same experimental setup in [26]. As shown in Table 3.1, Table 3.2, and Table 3.3, our method is comparable to some of the competitive discriminative dictionary learning algorithms such as SDL-D [69], SRSC [38], D-KSVD [24] and LC-KSVD [26]. Note that, our method is flexible enough that it can be applied over any dictionary learning schemes to enhance the discriminability.

3.5 Conclusion

We presented an information theoretic approach to dictionary learning that seeks a dictionary that is compact, reconstructive and discriminative for the task of image classification. The algorithm consists of dictionary selection and update stages. In the selection stage, an objective function is maximized using a greedy procedure to select a set of compact, reconstructive and discriminative atoms from an initial dictionary. In the update stage, a gradient ascent algorithm based on the quadratic mutual information is adopted to enhance the selected dictionary for improved reconstruction and discrimination. Both the proposed dictionary selection and update methods can be easily applied for other dictionary learning schemes.



(a) Four subjects (b) Sparse representation for images of four subjects (Sparsity = 3)



(c) Four digits (d) Sparse representation for four handwritten digits (Sparsity = 3)

Figure 3.1: Sparse representation using dictionaries learned by different approaches (SOMP [3], MMI-1 and MMI-2 [4]). For visualization, sparsity 3 is chosen, i.e., no more than three dictionary atoms are allowed in each sparse decomposition. When signals are represented at once as a linear combination of a common set of atoms, sparse coefficients of all the samples become points in the same coordinate space. Different classes are represented by different colors. The recognition accuracy is obtained through linear SVMs on the sparse coefficients. Our approach provides more discriminative sparse representation which leads to significantly better classification accuracy.

Input: Dictionary $\mathbf{D}^o$, signals $\mathbf{Y}$, class labels C, sparsity level T

Output: sparse coefficients $\mathbf{X}$, reconstruction $\hat{\mathbf{Y}}$

begin

Initialization stage:

1. Initialize $\mathbf{X}$ with any pursuit algorithm,

$i = 1, \cdots, N \quad \min_{\mathbf{x}_i} \|\mathbf{y}_i - \mathbf{D}^o\mathbf{x_i}\|_2^2 \; s.t. \; \|\mathbf{x}_i\|_0 \leq T.$

ITDS stage (shared atoms):

2. Estimate λ_1, λ_2 and λ_3 from $\mathbf{Y}$, $\mathbf{X}$ and C;

3. Find T most compact, discriminative and reconstructive atoms:

$\mathbf{D}^* \leftarrow \emptyset; \Gamma \leftarrow \emptyset$;

for *t=1 to T* **do**

$\mathbf{d}^* \leftarrow \arg \max_{\mathbf{d} \in \mathbf{D}^o \backslash \mathbf{D}^*} \lambda_1[I(\mathbf{D}^* \cup \mathbf{d}; \mathbf{D}^o \backslash (\mathbf{D}^* \cup \mathbf{d})) - I(\mathbf{D}^*; \mathbf{D}^o \backslash \mathbf{D}^*)] +$

$\lambda_2[I(\mathbf{X}_{\mathbf{D}^* \cup \mathbf{d}}; C) - I(\mathbf{X}_{\mathbf{D}^*}; C)] + \lambda_3[I(\mathbf{Y}; \mathbf{D}^* \cup \mathbf{d}) - I(\mathbf{Y}; \mathbf{D}^*)];$

$\mathbf{D}^* \leftarrow \mathbf{D}^* \bigcup \mathbf{d}^*;$

$\Gamma \leftarrow \Gamma \bigcup \gamma^*$, γ^* is the index of $\mathbf{d}^*$ in $\mathbf{D}^o$;

end

4. Compute sparse codes and reconstructions:

$\mathbf{X} \leftarrow pinv(\mathbf{D}^*)\mathbf{Y};$

$\hat{\mathbf{Y}} \leftarrow \mathbf{D}^*\mathbf{X};$

5. return $\mathbf{X}$, $\hat{\mathbf{Y}}$, $\mathbf{D}^*$, Γ ;

end

Algorithm 1: Sparse coding with global atoms.

Input: Dictionary $\mathbf{D}^o$, signals $\mathbf{Y} = \{\mathbf{Y}_c\}_{c=1}^p$, sparsity level T

Output: sparse coefficients $\{\mathbf{X}_c\}_{c=1}^p$, reconstruction $\{\hat{\mathbf{Y}}_c\}_{c=1}^p$

begin

Initialization stage:

1. Initialize $\mathbf{X}$ with any pursuit algorithm,

$i = 1, \cdots, N \quad \min_{\mathbf{x}_i} \|\mathbf{y}_i - \mathbf{D}^o\mathbf{x_i}\|_2^2 \; s.t. \; \|\mathbf{x}_i\|_0 \leq T.$

ITDS stage (dedicated atoms):

for *c=1 to p* **do**

2. $C_c \leftarrow \{c_i | c_i = 1$ if $y_i \in \mathbf{Y}_c$, 0 otherwise $\}$;

3. Estimate λ_1, λ_2 and λ_3 from $\mathbf{Y}_c$, $\mathbf{X}$ and C_c;

4. Find T most compact, discriminative and reconstructive atoms for class c:

$\mathbf{D}^* \leftarrow \emptyset$; $\Gamma \leftarrow \emptyset$;

for *t=1 to T* **do**

$\mathbf{d}^* \leftarrow \arg \max_{\mathbf{d} \in \mathbf{D}^o \backslash \mathbf{D}^*} \lambda_1 [I(\mathbf{D}^* \cup \mathbf{d}; \mathbf{D}^o \backslash (\mathbf{D}^* \cup \mathbf{d})) - I(\mathbf{D}^*; \mathbf{D}^o \backslash \mathbf{D}^*)] +$

$\lambda_2 [I(\mathbf{X}_{\mathbf{D}^* \cup \mathbf{d}}; C_c) - I(\mathbf{X}_{\mathbf{D}^*}; C_c)] + \lambda_3 [I(\mathbf{Y}_c; \mathbf{D}^* \cup \mathbf{d}) - I(\mathbf{Y}_c; \mathbf{D}^*)]$;

$\mathbf{D}^* \leftarrow \mathbf{D}^* \bigcup \mathbf{d}^*$;

$\Gamma \leftarrow \Gamma \bigcup \gamma^*$, γ^* is the index of $\mathbf{d}^*$ in $\mathbf{D}^o$;

end

$\mathbf{D}_c^* \leftarrow \mathbf{D}^*$; $\Gamma_c \leftarrow \Gamma$;

5. Compute sparse codes and reconstructions:

$\mathbf{X}_c \leftarrow pinv(\mathbf{D}_c^*)\mathbf{Y}_c$;

$\hat{\mathbf{Y}}_c \leftarrow \mathbf{D}_c^* \mathbf{X}_c$;

end

6. return $\{\mathbf{X}_c\}_{c=1}^p$, $\{\hat{\mathbf{Y}}_c\}_{c=1}^p$, $\{\mathbf{D}_c^*\}_{c=1}^p$, $\{\Gamma_c\}_{c=1}^p$;

end

Algorithm 2: Sparse coding with atoms per class.

Input: Dictionary $\mathbf{D}^o$, signals $\mathbf{Y} = \{\mathbf{Y}_c\}_{c=1}^p$, class labels C, sparsity level T, update step ν

Output: Learned dictionary $\mathbf{D}$, sparse coefficients $\mathbf{X}$, reconstruction $\hat{\mathbf{Y}}$

begin

Sparse coding stage:

Use supervised sparse coding to obtain $\{\mathbf{D}_c^*\}_{c=1}^p$.

ITDU stage:

foreach *class c* **do**

[In the shared atom case, use the global label C instead of C_c, and one iteration is required as the same $\mathbf{D}_c^*$ is used for all classes.]

$C_c \leftarrow \{c_i | c_i = 1 \text{ if } y_i \in \mathbf{Y}_c, 0 \text{ otherwise } \}$;

$\Phi_1 \leftarrow pinv(\mathbf{D}_c^*)^T$;

$\mathbf{X} \leftarrow pinv(\mathbf{D}_c^*)\mathbf{Y}$;

repeat

$\Phi_{k+1} = \Phi_k + \nu \frac{\partial I_Q(X, C_c)}{\partial \Phi}|_{\Phi=\Phi_k}$;

$\mathbf{D}^* \leftarrow pinv(\Phi_{k+1}^T)$;

$\mathbf{X} \leftarrow pinv(\mathbf{D}^*)\mathbf{Y}$;

until *convergence*;

$\mathbf{D}_c^* \leftarrow \mathbf{D}^*$;

end

foreach *class c* **do**

$\mathbf{X}_c \leftarrow pinv(\mathbf{D}_c^*)\mathbf{Y}_c$;

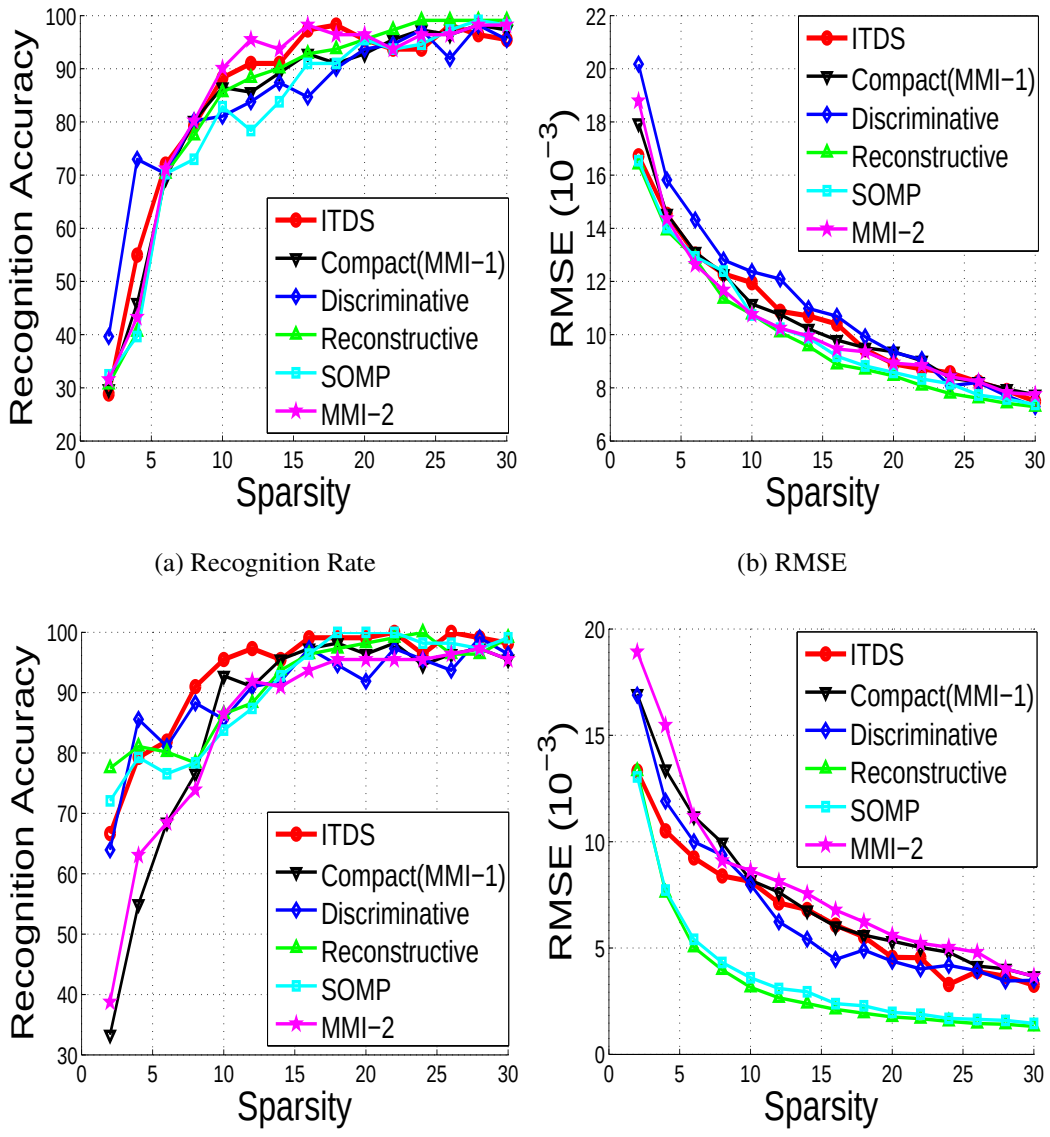
$\hat{\mathbf{Y}}_c \leftarrow \mathbf{D}_c^*\mathbf{X}_c$;

end

return $\{\mathbf{X}_c\}_{c=1}^p$, $\{\hat{\mathbf{Y}}_c\}_{c=1}^p$, $\{\mathbf{D}_c^*\}_{c=1}^p$;

end

Algorithm 3: Sparse coding with atom updates.



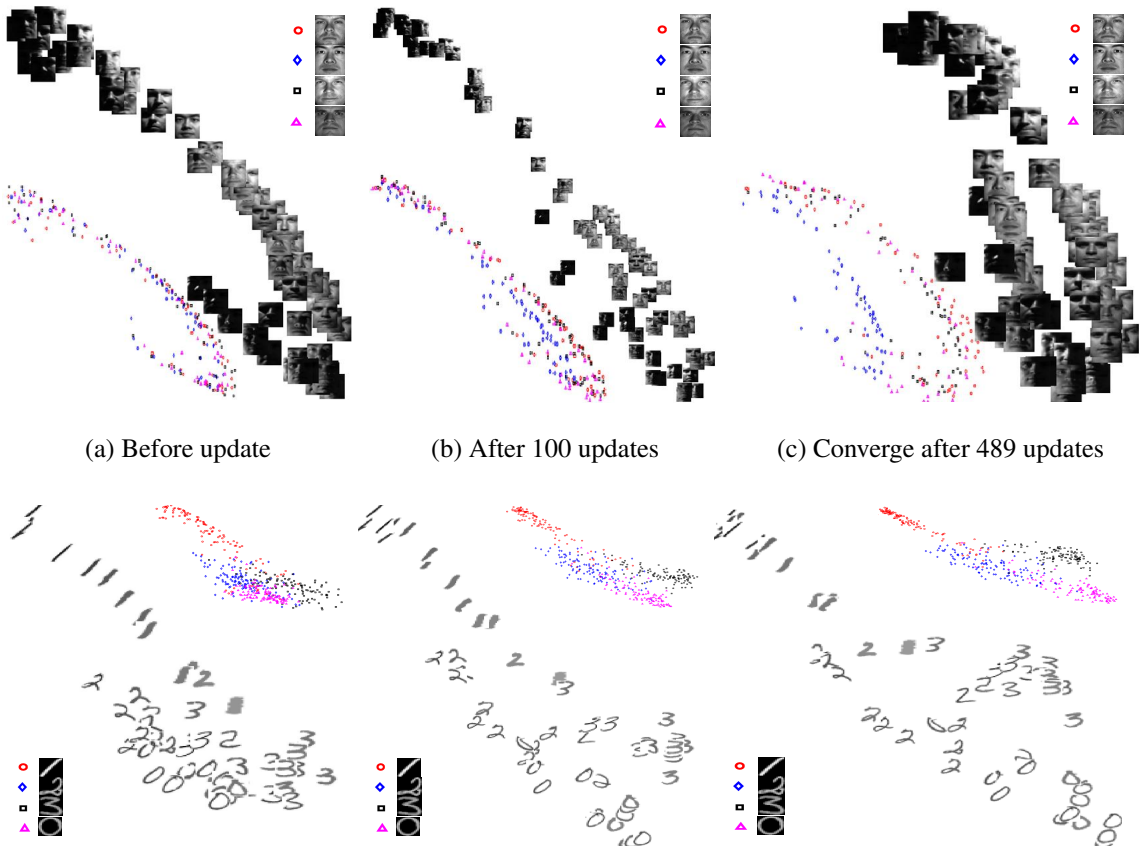
(a) Recognition Rate

(b) RMSE

(c) Recognition Rate

(d) RMSE

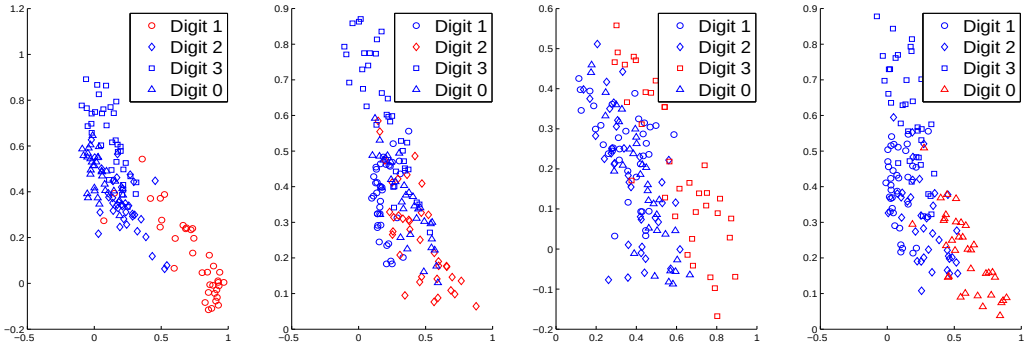
Figure 3.2: Recognition accuracy and RMSE on the YaleB dataset using different dictionary selection methods. We vary the sparsity level, i.e., the maximal number of dictionary atoms that are allowed in each sparse decomposition. In (a) and (b), a global set of common atoms are selected for all classes. In (c) and (d), a dedicated set of atoms are selected per class. In both cases, the proposed ITDS (red lines) provides the best recognition performance and moderate reconstruction error.



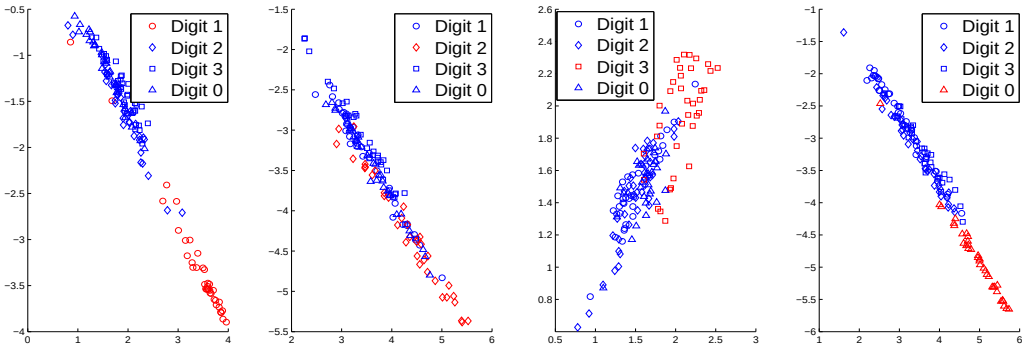
(a) Before update (b) After 100 updates (c) Converge after 489 updates

(d) Before update (e) After 50 updates (f) Converge after 171 updates

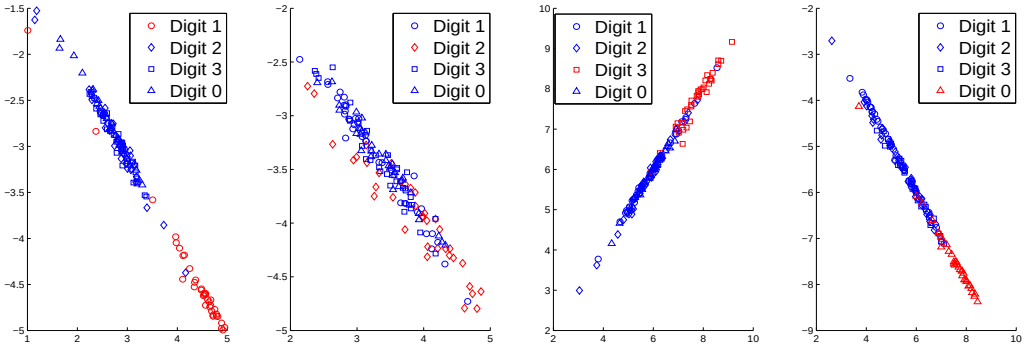
Figure 3.3: Information-theoretic dictionary update with global atoms shared over classes. For a better visual representation, sparsity 2 is chosen and a randomly selected subset of all samples are shown. The recognition rate associated with (a), (b), and (c) are: 30.63%, 42.34% and 51.35%. The recognition rate associated with (d), (e), and (f) are: 73.54%, 84.45% and 87.75%. Note that the proposed ITDU effectively enhances the discriminability of the set of common atoms.



(a) Before dictionary update (Acc.= 85.71%)

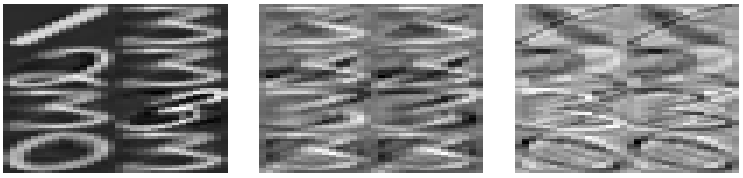


(b) After 30 update iterations (Acc.= 89.11%)

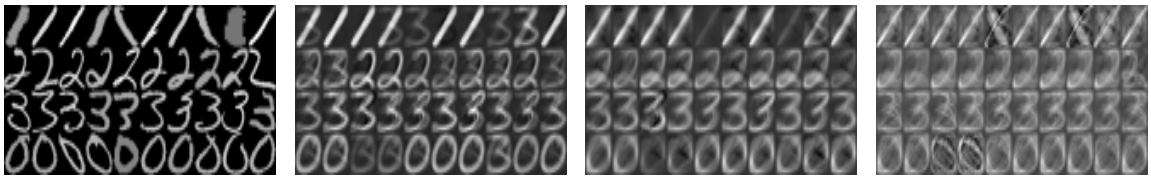


(c) Converge after 57 update iterations (Acc.= 90.47%)

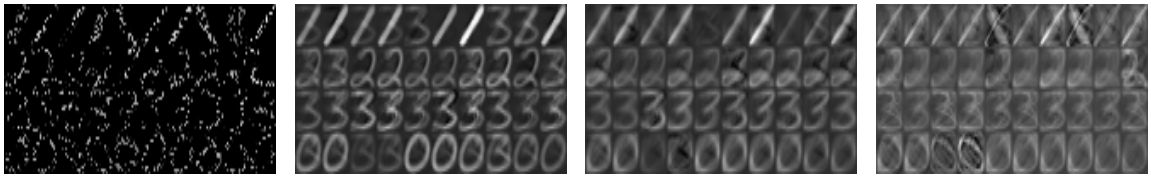
Figure 3.4: Information-theoretic dictionary update with dedicated atoms per class. The first four digits in the USPS digit dataset are used. Sparsity 2 is chosen for visualization. In each figure, signals are first represented at once as a linear combination of the dedicated atoms for the class colored by red, then sparse coefficients of all signals are plotted in the same 2-D coordinate space. The proposed ITDU effectively enhances the discriminability of the set of dedicated atoms.



(a) before update (b) 30 iterations (c) 57 iterations



(d) original images (e) before update (f) 30 iterations (g) 57 iterations



(h) noisy images (i) before update (j) 30 iterations (k) 57 iterations

Figure 3.5: Reconstruction using class dedicated atoms with the proposed dictionary update (sparsity 2 is used.). (a), (b) and (c) show the updated dictionary atoms, where from the top to the bottom the two atoms in each row are the dedicated atoms for class '1','2','3' and '0'. (e), (f) and (g) show the reconstruction to (d). (i), (j) and (k) show the reconstruction to (h). (h) are images in (d) with 60% missing pixels. Note that ITDU extracts the common internal structure of each class and eliminates the variation within the class, which leads to more accurate classification.

Table 3.1: Classification rate (%) on the USPS dataset.

Proposed	SDL-D [69]	SRSC [38]	FDDL [65]	k-NN	SVM-Gauss
98.28	96.44	93.95	96.31	94.80	95.80

Table 3.2: Classification rate (%) on the 15 scenes dataset.

Proposed	ScSPM [77]	KSPM [76]	KC [78]	LSPM [77]
81.13	80.28	76.73	76.67	65.32

Table 3.3: Classification rate (%) on the Extended YaleB face dataset.

Proposed	D-KSVD [24]	LC-KSVD [26]	K-SVD [20]	SRC [79]	LLC [80]
95.39	94.10	95.00	93.1	80.5	90.7

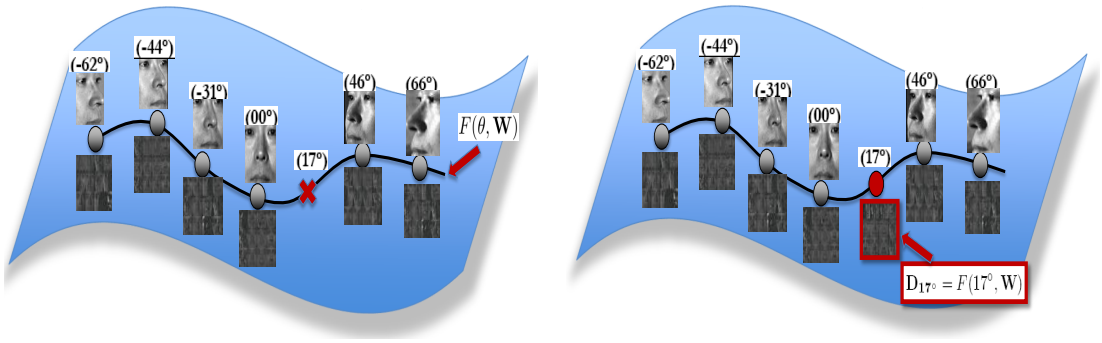
Chapter 4

Domain Adaptive Dictionary Learning

4.1 Introduction

In recent years, sparse and redundant modeling of signals has received a lot of attention from the vision community [56]. This is mainly due to the fact that signals or images of interest are sparse or compressible in some dictionary. In other words, they can be well approximated by a linear combination of a few elements (also known as atoms) of a redundant dictionary. This dictionary can either be an analytic dictionary such as wavelets or it can be directly trained from data. It has been observed that dictionaries learned directly from data provide better representation and hence can improve the performance of many applications such as image restoration and classification [55].

When designing dictionaries for image classification tasks, we are often confronted with situations where conditions in the training set are different from those present during testing. For example, in the case of face recognition, more than one familiar view may be available for training. Such training faces may be obtained from a live or recorded video sequences, where a range of views are observed. However, the test images can contain conditions that are not necessarily presented in the training images such as a face in a different pose. The problem of transforming a dictionary trained from one visual domain to another without changing signal sparse representations can be viewed as a problem of domain adaptation [16] and transfer learning [17].



(a) Example dictionaries learned at known poses with observations.

(b) Domain adapted dictionary at a pose ($\theta = 17^\circ$) associated with no observations.

Figure 4.1: Overview of our approach. Consider example dictionaries corresponding to faces at different azimuths. (a) shows a depiction of example dictionaries over a curve on a dictionary manifold which will be discussed later. Given example dictionaries, our approach learns the underlying dictionary function $F(\theta, \mathbf{W})$. In (b), the dictionary corresponding to a domain associated with observations is obtained by evaluating the learned dictionary function at the corresponding domain parameters.

Given the same set of signals observed in different visual domains, our goal is to learn a dictionary for the new domain without corresponding observations. We formulate this problem of dictionary transformation in a function learning framework, i.e., dictionaries across different domains are modeled by a parametric function. The dictionary function parameters and domain-invariant sparse codes are then jointly learned by solving an optimization problem. As shown in Figure 4.1, given a learned dictionary function, a dictionary adapted to a new domain is obtained by evaluating such a dictionary function at the corresponding domain parameters, e.g., pose angles.

For the case of view variations, linear interpolation methods have been discussed

in [81] to predict intermediate views of faces given a frontal and profile views. These methods essentially apply linear regression on the PCA coefficients corresponding to two different views. In [82], Vetter and Poggio present a method for learning linear transformations from a basis set of prototypical views. Their approach is based on the linear class property which essentially states that if a 3D view of an object can be represented as the weighted sum of views of other objects, its rotated view is a linear combination of the rotated views of the other objects with the same weights [82], [83], [84]. Note that our method is more general than the above mentioned methods in that it is applicable to visual domains other than pose. Second, our method is designed to maintain consistent sparse coefficients for the same signal observed in different domains. Furthermore, our method is based on the recent dictionary learning methods and is able to learn dictionaries that are more general than the ones resulting from PCA.

This chapter makes the following contributions:

- A general continuous function learning framework is presented for the task of dictionary transformations across domains.
- A simple and efficient optimization procedure is presented that learns dictionary function parameters and domain-invariant sparse codes simultaneously.
- Experiments for various applications, including pose alignment, pose and illumination estimation and face recognition across pose, are presented.

4.2 Overall Approach

We consider the problem of dictionary transformations in a learning framework, where we are provided with a few examples of dictionaries $\mathbf{D}_i$ with corresponding domain parameter θ_i. Let the parameter space be denoted by Θ, i.e. $\theta_i \in \Theta$. Let the *dictionary space* be denoted $\mathcal{D}$. The problem then boils down to constructing a mapping function $F : \Theta \mapsto \mathcal{D}$. In the simple case where $\Theta = \mathbb{R}$ and $\mathcal{D} = \mathbb{R}^n$, the problem of fitting a function can be solved efficiently using curve fitting techniques [85]. A dictionary of d atoms in $\mathbb{R}^n$ is often considered as an $n \times d$ real matrix or equivalently a point in $\mathbb{R}^{n \times d}$. However, often times there are additional constraints on dictionaries that make the identification with $\mathbb{R}^{n \times d}$ not well-motivated. We present below a few such constraints:

- Subspaces: For the special case of under-complete dictionaries where the matrix is full-rank and thus represents a choice of basis vectors for a d-dimensional subspace in $\mathbb{R}^n$, the dictionary space is naturally considered as a Grassmann manifold $\mathcal{G}_{n,d}$ [86]. The geometry of the Grassmann manifold is studied either as a quotient-space of the special orthogonal group or in terms of full-rank projection matrices, both of which result in non-Euclidean geometric structures.

- Products of subspaces: In many cases, it is convenient to think of the dictionary as a union of subspaces, e.g. a line and a plane. This structure has been utilized in many applications such as generalized PCA (GPCA), sparse subspace clustering [87] etc. In this case, the dictionary-space becomes a subset of the product space of Grassmann manifolds.

- Overcomplete dictionaries: In the most general case one considers an over-complete set of basis vectors, where each basis vector has unit-norm, i.e. each basis vector is a point on the hypershere $\mathbb{S}^{n-1}$. In this case, the dictionary space becomes a subset of the product-space $\mathbb{S}^{(n-1)\times d}$.

To extend classic multi-variate function fitting to manifolds such as the ones above, one needs additional differential geometric tools. In our case, we propose extrinsic approaches that rely on embedding the manifold into an ambient vector space, perform function/curve fitting in the ambient space, and project the results back to the manifold of interest. This is conceptually simpler, and we find in our experiments that this approach works very well for the problems under consideration. The choice of embedding is in general not unique. We describe below the embedding and the corresponding projection operations for the manifolds of interest describe above.

- Subspaces: Each point in $\mathcal{G}_{n,d}$ corresponds to a d-dimensional subspace of $\mathbb{R}^n$. Given a choice of orthonormal basis vectors for the subspace $\mathbb{Y}$, the $n \times n$ projection matrix given by $\mathbf{P} = \mathbb{Y}\mathbb{Y}^T$ is a unique representation for the subspace. The projection matrix represntation can then be embedded into the ambient vector-space $\mathbb{R}^{n\times n}$. The projection operation $\mathbf{\Pi}$ is given by $\mathbf{\Pi}(\mathbf{M}) = \mathbf{U}\mathbf{U}^\mathbf{T}$, where $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathbf{T}$ is a rank-d SVD of $\mathbf{M}$ [5].

- Products of subspaces: Following the procedure above, each component of the product space can be embedded into a different vector-space and the projected back to the manifold using the corresponding projection operation.

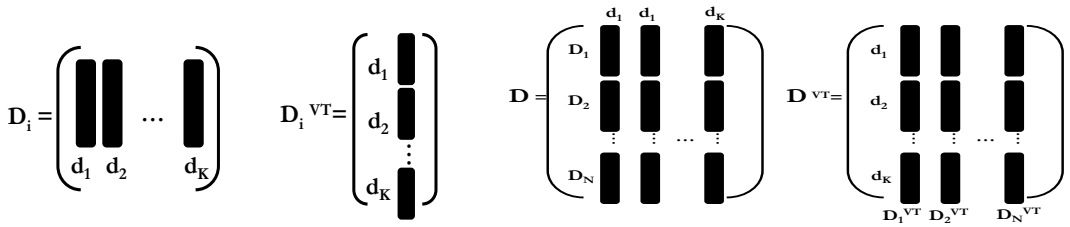
- Overcomplete dictionaries: The embedding from $\mathbb{S}^{n-1}$ to $\mathbb{R}^n$ is given by a vectorial representation with unit-norm. The projection $\mathbf{\Pi} : \mathbb{R}^n \mapsto \mathbb{S}^{n-1}$ is given by $\mathbf{\Pi}(\mathbf{V}) = \frac{\mathbf{V}}{\|\mathbf{V}\|}$, where $\|.\|$ is the standard Euclidean norm. A similar operation on the product-space $\mathbb{S}^{(n-1)\times d}$ can be defined by component-wise projection operations.

In specific examples in the chapter, we consider the case of over-complete dictionaries. We adopt the embedding and projection approach described above as a means to exploit the wealth of function-fitting techniques available for vector-spaces. Next, we describe the technique we adopt.

4.2.1 Problem Formulation

We denote the same set of P signals observed in N different domains as $\{\mathbf{Y_1}, ..., \mathbf{Y_N}\}$, where $\mathbf{Y_i} = [\mathbf{y_{i1}}, ..., \mathbf{y_{iP}}],\ \mathbf{y_{ip}} \in \mathbb{R}^n$. Thus, $\mathbf{y_{ip}}$ denotes the p^{th} signal observed in the i^{th} domain. In the following, we will use $\mathbf{D_i}$ as the vector-space embedded dictionary. Let $\mathbf{D_i}$ denote the dictionary for the i^{th} domain, where $\mathbf{D_i} = [\mathbf{d_{i1}}...\mathbf{d_{iK}}],\ \mathbf{d_{ik}} \in \mathbb{R}^n$. We define a *vector transpose* (VT) operation over dictionaries as illustrated in Figure 4.2. The VT operator treats each individual dictionary atom as a value and then perform the typical matrix transpose operation. Let $\mathbf{D}$ denote the stack dictionary shown in Figure 4.2b over all N domains. It is noted that $\mathbf{D} = [\mathbf{D^{VT}}]^{\mathbf{VT}}$.

The domain dictionary learning problem can be formulated as (4.1). Let $\mathbf{X} = [\mathbf{x_1}, ..., \mathbf{x_P}],\ \mathbf{x_p} \in \mathbb{R}^K$, be the sparse code matrix. The set of domain dictionary $\{\mathbf{D_i}\}_i^N$ learned through (4.1) enable the same sparse codes $\mathbf{x_p}$ for a signal $\mathbf{y_p}$ observed across N



(a) $\mathbf{D_i}$ vs. $\mathbf{D_i}^{VT}$ (b) $\mathbf{D}$ vs. $\mathbf{D}^{VT}$

Figure 4.2: The vector transpose (VT) operator over dictionaries.

different domains to achieve domain adaptation.

$$\underset{\{\mathbf{D_i}\}_\mathbf{i}^\mathbf{N},\mathbf{X}}{\arg\min} \sum_{i}^{N} \|\mathbf{Y_i} - \mathbf{D_i}\mathbf{X}\|_F^2 \;\; s.t.\; \forall p \; \|\mathbf{x}_p\|_o \leq T, \tag{4.1}$$

where $\|\mathbf{x}\|_o$ counts the number of non-zero values in $\mathbf{x}$. T is a sparsity constant.

We propose to model domain dictionaries $\mathbf{D_i}$ through a parametric function in (4.2), where $\boldsymbol{\theta_i}$ denotes a vector of domain parameters, e.g., view point angles, illumination conditions, etc., and $\mathbf{W}$ denotes the dictionary function parameters.

$$\mathbf{D_i} = F(\boldsymbol{\theta_i}, \mathbf{W}) \tag{4.2}$$

Applying (4.2) to (4.1), we formulate the domain dictionary function learning as (4.3).

$$\underset{\mathbf{W},\mathbf{X}}{\arg\min} \sum_{i}^{N} \|\mathbf{Y_i} - F(\boldsymbol{\theta_i}, \mathbf{W})\mathbf{X}\|_F^2 \;\; s.t.\; \forall p \; \|\mathbf{x}_p\|_o \leq T. \tag{4.3}$$

Once a dictionary is estimated it is projected back to the dictionary-space by the projection operation described earlier.

4.2.2 Domain Dictionary Function Learning

We first adopt power polynomials to model $\mathbf{D_i^{VT}}$ in Figure 4.2a through the following dictionary function $F(\boldsymbol{\theta_i}, \mathbf{W})$,

$$F(\boldsymbol{\theta_i}, \mathbf{W}) = w_0 + \sum_{s=1}^{S} w_{1s}\theta_{is} + ... + \sum_{s=1}^{S} w_{ms}\theta_{is}^{m} \tag{4.4}$$

where we assume S-dimensional domain parameter vectors and an m^{th}-degree polynomial model. For example, given $\boldsymbol{\theta_i}$ a 2-dimensional domain parameter vector, a quadratic dictionary function is defined as,

$$F(\boldsymbol{\theta_i}, \mathbf{W}) = w_0 + w_{11}\theta_{i1} + w_{12}\theta_{i2} + w_{21}\theta_{i1}^2 + w_{22}\theta_{i2}^2$$

Given $\mathbf{D_i}$ contains K atoms and each dictionary atom is in the $\mathbb{R}^n$ space, as $\mathbf{D_i^{VT}} = F(\boldsymbol{\theta_i}, \mathbf{W})$, it can be noted from Figure 4.2 that w_{ms} is a nK-sized vector. We define the function parameter matrix $\mathbf{W}$ and the domain parameter matrix $\boldsymbol{\Theta}$ as

$$\mathbf{W} = \begin{bmatrix} w_0^{(1)} & w_0^{(2)} & w_0^{(3)} & ... & w_0^{(nK)} \\ w_{11}^{(1)} & w_{11}^{(2)} & w_{11}^{(3)} & ... & w_{11}^{(nK)} \\ & & & . & \\ & & & . & \\ & & & . & \\ w_{mS}^{(1)} & w_{mS}^{(2)} & w_{mS}^{(3)} & ... & w_{mS}^{(nK)} \end{bmatrix} \qquad \boldsymbol{\Theta} = \begin{bmatrix} 1 & 1 & 1 & ... & 1 \\ \theta_{11} & \theta_{21} & \theta_{31} & ... & \theta_{N1} \\ & & & . & \\ & & & . & \\ & & & . & \\ \theta_{1S}^m & \theta_{2S}^m & \theta_{3S}^m & ... & \theta_{NS}^m \end{bmatrix}$$

Each row of $\mathbf{W}$ corresponds to the nK-sized w_{ms}^T, and $\mathbf{W} \in \mathbb{R}^{(mS+1)\times nK}$. N different domains are assumed and $\boldsymbol{\Theta} \in \mathbb{R}^{(mS+1)\times N}$. With the matrix $\mathbf{W}$ and $\boldsymbol{\Theta}$, (4.4) can be written

as,

$$\mathbf{D^{VT}} = \mathbf{W^{T}\Theta} \tag{4.5}$$

where $\mathbf{D^{VT}}$ is defined in Figure 4.2b. Now dictionary function learning formulated in (4.3) can be written as,

$$\underset{\mathbf{W},\mathbf{X}}{\arg\min} \|\mathbf{Y} - [\mathbf{W^{T}\Theta}]^{\mathbf{VT}}\mathbf{X}\|_F^2 \;\; s.t.\; \forall p \; \|\mathbf{x}_p\|_o \leq T \tag{4.6}$$

where $\mathbf{Y}$ is the stacked training signals observed in different domains as illustrated in Figure 4.3. With the objective function defined in (4.6), the dictionary function learning can be performed in the following steps,

Step 1: Obtain the sparse coefficients X and $[\mathbf{W^{T}\Theta}]^{\mathbf{VT}}$ via any dictionary learning method, e.g., K-SVD [20].

Step 2: Given the domain parameter matrix $\mathbf{\Theta}$, the optimal dictionary function can be obtained as [88],

$$\mathbf{W} = [\mathbf{\Theta\Theta^{T}}]^{-1}\mathbf{\Theta}[[[\mathbf{W^{T}\Theta}]^{\mathbf{VT}}]^{\mathbf{VT}}]^{\mathbf{T}}. \tag{4.7}$$

Step 3: Sample the dictionary function at desired parameters values, and project it to the dictionary-space using an appropriate projection operation.

4.2.3 Non-linear Dictionary Function Models

Till now, we only assume power polynomials for the dictionary model. In this section, we discuss non-linear dictionary functions. We only focus on linearizeable functions, and a general Newton's method based approach to learn a non-linear dictionary function is briefly discussed in Algorithm 4..

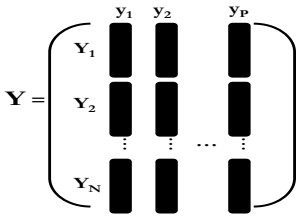


Figure 4.3: The stack P training signals observed in N different domains.

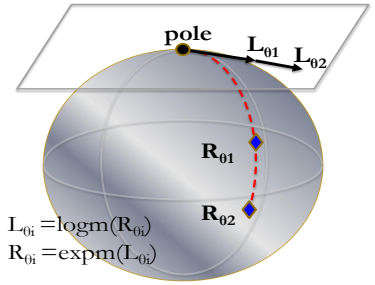


Figure 4.4: Illustration of exponential maps *expm* and inverse exponential maps *logm* [5].

4.2.3.1 Linearizeable Models

There are several well-known linearizeable models, such as the Cobb-Douglass model, the logistic model, etc. We use the Cobb-Douglass model as the example to discuss in detail how dictionary function learning can be performed over these linearizable models.

The Cobb-Douglass model is written as,

$$\mathbf{D_i^{VT}} = F(\boldsymbol{\theta_i}, \mathbf{W}) = w_0 \exp(\sum_{s=1}^{S} w_{1s}\theta_{is} + ... + \sum_{s=1}^{S} w_{ms}\theta_{is}^m) \tag{4.8}$$

The logarithmic transformation yields,

$$\log(\mathbf{D_i^{VT}}) = \log(w_0) + \sum_{s=1}^{S} w_{1s}\theta_{is} + ... + \sum_{s=1}^{S} w_{ms}\theta_{is}^m$$

As the right side of (4.8) is in the same linear form as (4.4), we can define the corresponding function parameter matrix $\mathbf{W}$ and the domain parameter matrix $\boldsymbol{\Theta}$ as discussed. The dictionary function learning is written as,

$$\arg\min_{\mathbf{W},\mathbf{X}} \|\mathbf{Y} - [\exp(\mathbf{W^T\Theta})]^{\mathbf{VT}}\mathbf{X}\|_F^2 \;\; s.t.\; \forall p \; \|\mathbf{x}_p\|_o \leq T.$$

Through any dictionary learning methods, we obtain $[[\exp(\mathbf{W^T\Theta})]^{\mathbf{T}}]^{\mathbf{VT}}$ and $\mathbf{X}$.

Then, the dictionary function is obtained as,

$$\mathbf{W} = [\mathbf{\Theta}\mathbf{\Theta}^{\mathbf{T}}]^{-\mathbf{1}}\mathbf{\Theta}[\log([[\exp(\mathbf{W}^{\mathbf{T}}\mathbf{\Theta})]^{\mathbf{VT}}]^{\mathbf{VT}})]^{\mathbf{T}}.$$

Input: signals in N different domains $\{\mathbf{Y_i}\}_{\mathbf{i=1}}^{\mathbf{N}}$, domain parameter matrix $\mathbf{\Theta}$

Output: dictionary function $\mathbf{W}$

begin

Initialization:

1. Create the stack signal $\mathbf{Y}$ and initialize $\mathbf{D}$ from $\mathbf{Y}$ using K-SVD;

2. Initialize $\mathbf{W}$ with random values ;

repeat

3. Compute current residuals: $\mathbf{R} \leftarrow \mathbf{D} - \mathbf{F}(\mathbf{\Theta}, \mathbf{W})$;

4. Compute the row vector of derivatives w.r.t. $\mathbf{W}$ evaluated at $\mathbf{\Theta}$

$\mathbf{P} \leftarrow \nabla\mathbf{F}(\mathbf{\Theta}, \mathbf{W})$;

5. Learn the linear dictionary function $\mathbf{B}$ using $\mathbf{R} = \mathbf{PB}$

6. Update the dictionary function parameters: $\mathbf{W} \leftarrow \mathbf{W} + \lambda\mathbf{B}$

until *convergence*;

7. return $\mathbf{W}$;

end

Algorithm 4: A general method for nonlinear dictionary function learning.

4.2.4 Domain Parameter Estimation

Given a learned dictionary function $F(\boldsymbol{\theta}, \mathbf{W})$, the domain parameters $\boldsymbol{\theta_y}$ associated with an unknown image $\mathbf{y}$, e.g., pose (azimuth, altitude) or light source directions (azimuth, altitude), can be estimated using Algorithm 5.

It is noted that we adopt the following strategy to represent the domain parameter vector $\boldsymbol{\theta}$ for each pose in a linear space: we first obtain the rotation matrix $\mathbf{R}_\theta$ from the az-

imuth and altitude of a pose; we then compute the inverse exponential map of the rotation matrix $\text{logm}(\mathbf{R}_\theta)$ as shown in Figure 4.4. We denote $\boldsymbol{\theta}$ using the upper triangular part of the resulting skew-symmetric matrix [5]. The exponential map operation in Figure 4.4 is used to recover the azimuth and altitude from estimated domain parameters. We represent light source directions in the same way.

4.3 Experimental Evaluation

We conduct our experiments using two public face datasets: the CMU PIE dataset [89] and the Extended YaleB dataset [90]. The CMU PIE dataset consists of 68 subjects in 13 poses and 21 lighting conditions. In our experiments we use 9 poses which have approximately the same camera altitude, as shown in the first row of Figure 4.5. The Extended YaleB dataset consists of 38 subjects in 64 lighting conditions. All images are in 64×48 size. We will first evaluate the basic behaviors of dictionary functions through pose alignment. Then we will demonstrate the effectiveness of dictionary functions in face recognition and domain estimation.

4.3.1 Dictionary Functions for Pose alignment

4.3.1.1 Frontal Face Alignment

In Figure 4.5, we align different face poses to the frontal view. We learn for each subject in the PIE dataset a linear dictionary function $F(\boldsymbol{\theta}, \mathbf{W})$ (m=4) using 5 out of 9 poses. The training poses are highlighted in blue in the first row of Figure 4.5. Given a source image $\mathbf{y_s}$, we first estimate the domain parameters $\boldsymbol{\theta_s}$, i.e., the pose azimuth here, by

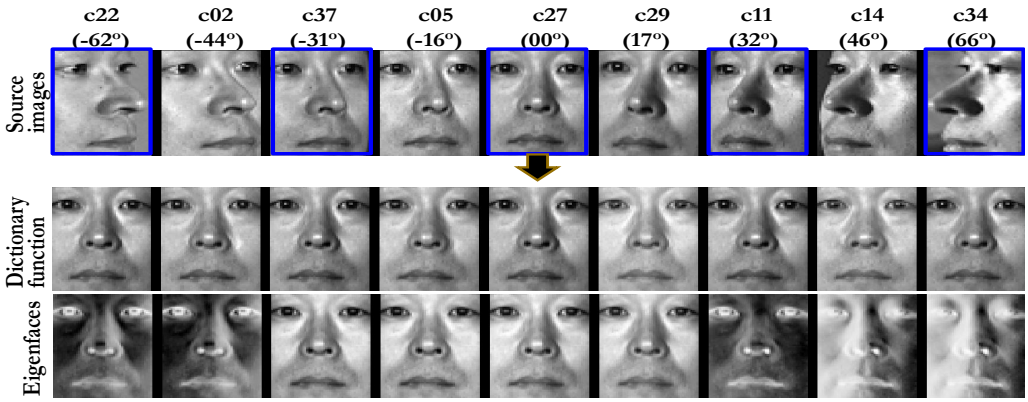
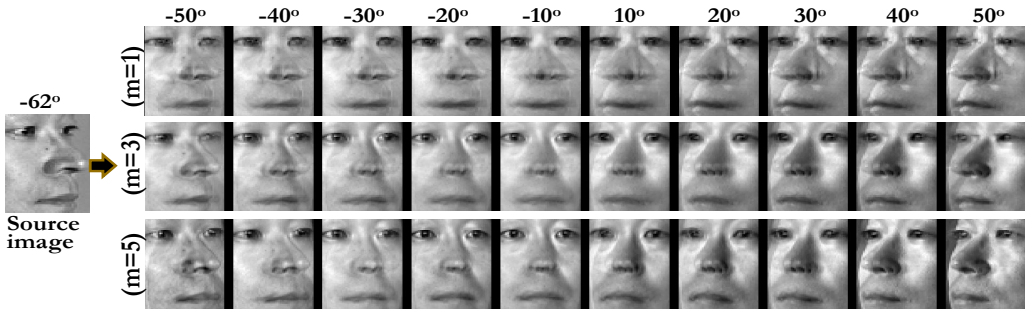


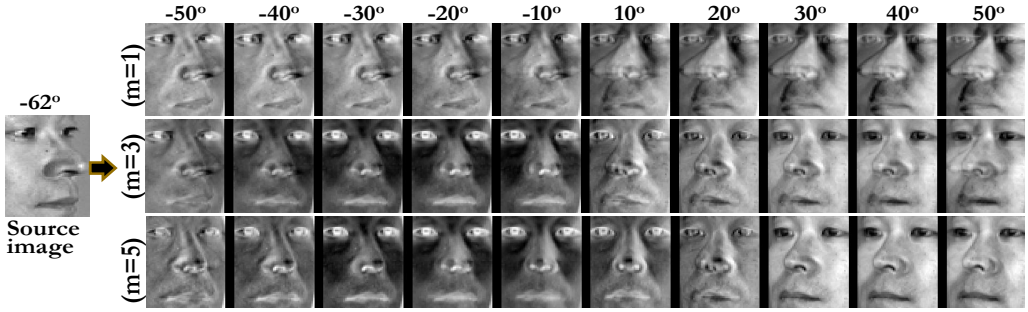
Figure 4.5: Frontal face alignment. For the first row of source images, pose azimuths are shown below the camera numbers. Poses highlighted in blue are known poses to learn a linear dictionary function (m=4), and the remaining are unknown poses. The second and third rows show the aligned face to each corresponding source image using the linear dictionary function and Eigenfaces respectively.

following Algorithm 5. We then obtain the sparse representation $\mathbf{x_s}$ of the source image as $\min_{\mathbf{x_s}} \|\mathbf{y_s} - F(\boldsymbol{\theta_s}, \mathbf{W})\mathbf{x_s}\|_2^2,\ s.t.\ \|\mathbf{x_s}\|_o \leq T$ (sparsity level) using any pursuit methods such as OMP [60]. We specify the fontal pose azimuth (00^o) as the parameter for the target domain $\boldsymbol{\theta_t}$, and obtain the frontal view image $\mathbf{y_t}$ as $\mathbf{y_t} = F(\boldsymbol{\theta_t}, \mathbf{W})\mathbf{x_s}$. The second row of Figure 4.5 shows the aligned frontal view images to the respective poses in the first row. These aligned frontal faces are close to the actual image, i.e., c27 in the first row. It is noted that images with poses c02, c05, c29 and c14 are unknown poses to the learned dictionary function.

For comparison, we learn Eigenfaces for each of the 5 training poses and obtain adapted Eigenfaces at 4 unknown poses using the same function fitting method in our framework. We then project each source image (mean-subtracted) on the respective eigne-



(a) Pose synthesis using a linear dictionary function



(b) Pose synthesis using Eigenfaces

Figure 4.6: Pose synthesis using various degrees of dictionary polynomials. All the synthesized poses are unknown to learned dictionary functions and associated with no actual observations. m is the degree of a dictionary polynomial in (4.4).

faces and use frontal Eigenfaces to reconstruct the aligned image shown in the third row of Figure 4.5. The proposed method of jointly learning the dictionary function parameters and domain-invariant sparse codes in (4.6) significantly outperforms the Eigenfaces approach, which fails for large pose variations.

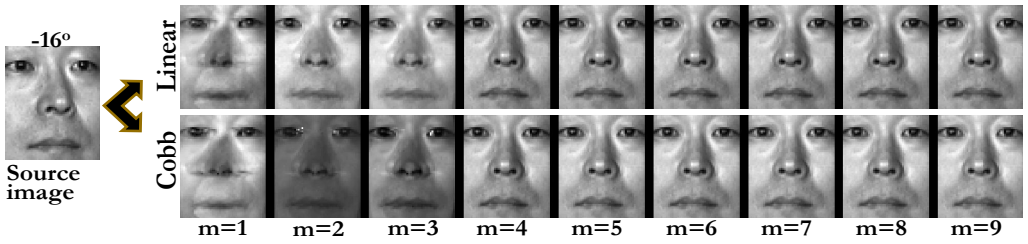
4.3.1.2 Pose Synthesis

In Figure 4.6, we synthesize new poses at any given pose azimuth. We learn for each subject in the PIE dataset a linear dictionary function $F(\boldsymbol{\theta}, \mathbf{W})$ using all 9 poses. In

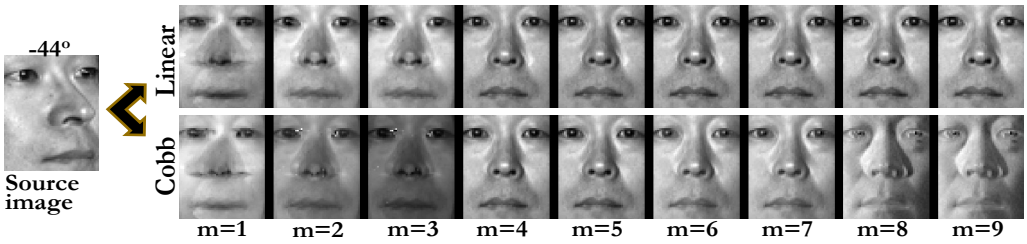
Figure 4.6a, given a source image $\mathbf{y_s}$ in a profile pose (-62^o), we first estimate the domain parameters $\boldsymbol{\theta_s}$ for the source image, and sparsely decompose it over $F(\boldsymbol{\theta_s}, \mathbf{W})$ for its sparse representation $\mathbf{x_s}$. We specify every 10^o pose azimuth in $[-50^o, 50^o]$ as parameters for the target domain $\boldsymbol{\theta_t}$, and obtain a synthesized pose image $\mathbf{y_t}$ as $\mathbf{y_t} = F(\boldsymbol{\theta_t}, \mathbf{W})\mathbf{x_s}$. It is noted that none of the target poses are associated with actual observations. As shown in Figure 4.6a, we obtain reasonable synthesized images at poses with no observations. We observe improved synthesis performance by increasing the value of m, i.e., the degree of a dictionary polynomial. In Figure 4.6b, we perform curve fitting over Eigenfaces as discussed. The proposed dictionary function learning framework exhibits better synthesis performance.

4.3.1.3 Linear vs. Non-linear

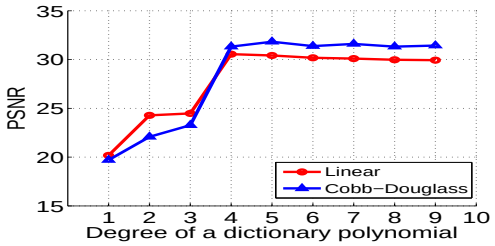
In Figure 4.7, we conduct the same frontal face alignment experiments discussed above. Now we learn for each subject both a linear and a nonlinear Cobb-Douglass dictionary function discussed in Section 4.2.3. As a Cobb-Douglass function is linearizeable, various degrees of polynomials are experimented for both linear and nonlinear dictionary function learning. As shown in Figure 4.7a and Figure 4.7c, the nonlinear Cobb-Douglass dictionary function exhibits better reconstruction while aligning pose c05, which is also indicated by the higher PSNR values. However, in Figure 4.7b and 4.7d, we notice that the Cobb-Douglass dictionary function exhibits better alignment performance only when $m \leq 7$, and then the performance drops dramatically. Therefore, a linear dictionary function is a more robust choice over a nonlinear Cobb-Douglass dictionary function;



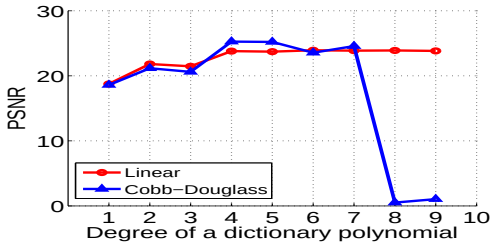
(a) Pose c05 frontal alignment



(b) Pose c02 frontal alignment



(c) Pose c05 alignment PSNR



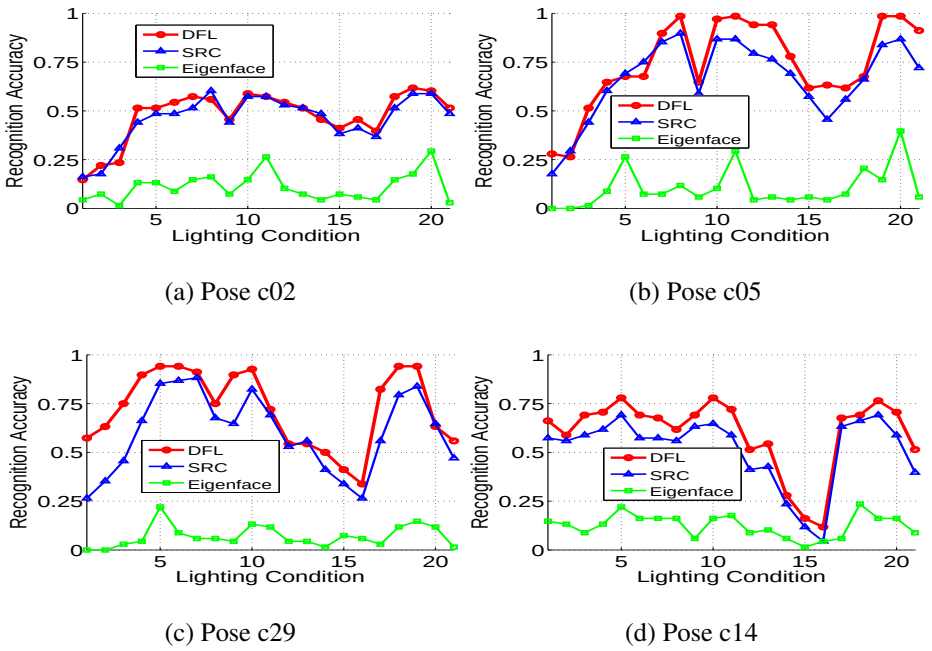
(d) Pose c02 frontal PSNR

Figure 4.7: Linear vs. non-linear dictionary functions. m is the degree of a dictionary polynomial in (4.4) and (4.8) .

however, at proper configurations, a nonlinear Cobb-Douglass dictionary function outperforms a linear dictionary function.

4.3.2 Dictionary Functions for Classification

Two face recognition methods are adopted for comparisons: Eigenfaces [91] and SRC [22]. Eigenfaces is a benchmark algorithm for face recognition. SRC is a state of the art



(a) Pose c02

(b) Pose c05

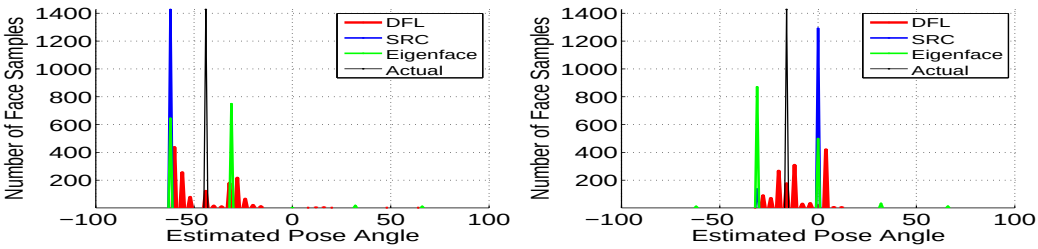
(c) Pose c29

(d) Pose c14

Figure 4.8: Face recognition accuracy on the CMU PIE dataset. The proposed method is denoted as DFL in color red.

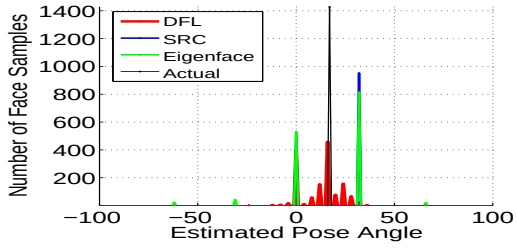
method to use sparse representation for face recognition. We denote our method as the Dictionary Function Learning (DFL) method. For a fair comparison, we adopt exactly the same configurations for all three methods, i.e., we use 68 subjects in 5 poses c22, c37, c27, c11 and c34 in the PIE dataset for training, and the remaining 4 poses for testing.

For the SRC method, we form a dictionary from the training data for each pose of a subject. For the proposed DFL method, we learn from the training data a dictionary function across pose for each subject. In SRC and DFL, a testing image is classified using the subject label associated with the dictionary or the dictionary function respectively that gives the minimal reconstruction error. In Eigenfaces, a nearest neighbor classifier is used. In Figure 4.8, we present the face recognition accuracy on the PIE dataset for different testing poses under each lighting condition. The proposed DFL method outperforms both

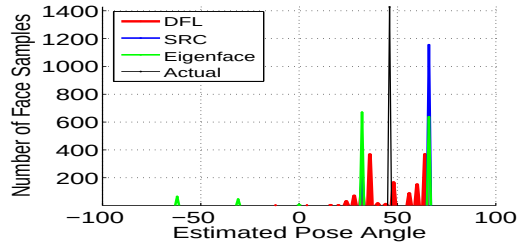


(a) Pose c02 (-44^o)

(b) Pose c05 (-16^o)



(c) Pose c29 (17^o)



(d) Pose c14 (46^o)

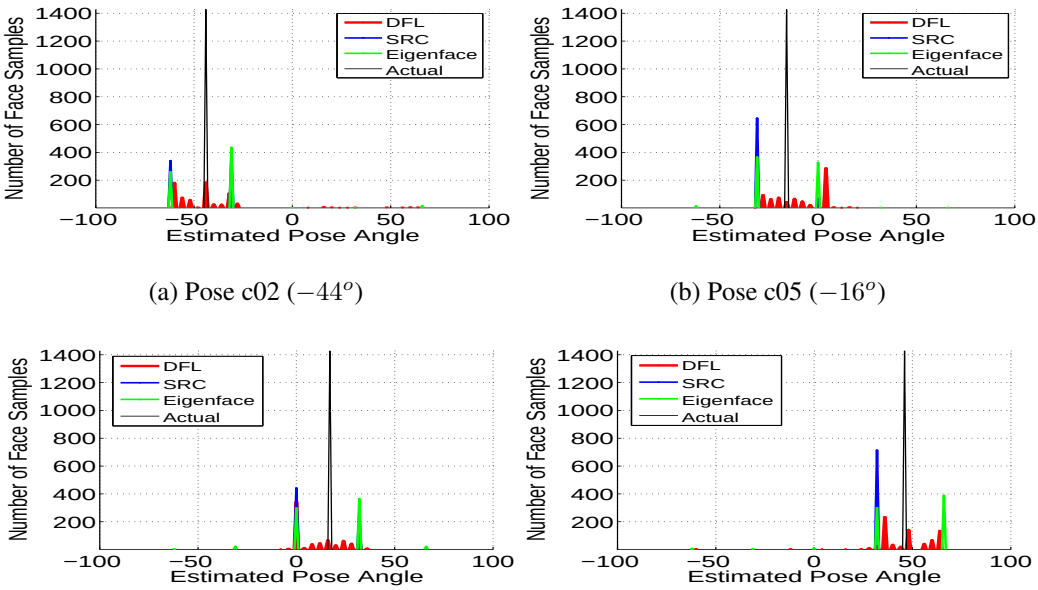
Figure 4.9: Pose azimuth estimation histogram (*known* subjects). Azimuths estimated using the proposed dictionary functions (red) spread around the true values (black).

Eigenfaces and SRC methods for all testing poses.

4.3.3 Dictionary Functions for Domain Estimation

4.3.3.1 Pose Estimation

As described in Algorithm 5, given a dictionary function, we can estimate the domain parameters associated with an unknown image, e.g., view point or illumination. It can be observed from the face recognition experiments discussed above that the SRC and eigenfaces methods can also estimate the domain parameters based on the domain associated with each dictionary or each training sample. However, the domain estimation accuracy using such recognition methods is limited by the domain discretization steps present in



(a) Pose c02 (-44^o)

(b) Pose c05 (-16^o)

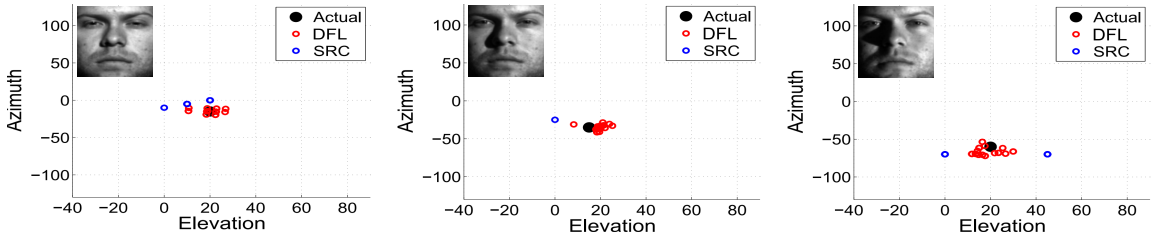
(c) Pose c29 (17^o)

(d) Pose c14 (46^o)

Figure 4.10: Pose azimuth estimation histogram (*unknown* subjects). Azimuths estimated using the proposed dictionary functions (red) spread around the true values (black).

the training data. We perform pose estimation along with the classification experiments above. We have 4 testing poses and each pose contains 1428 images (68 subjects in 21 lighting conditions). Figure 4.9 shows the histogram of pose azimuth estimation. We notice that poses estimated from Eigenfaces and SRC methods are limited to one of the 5 training pose azimuths, i.e., -62^o (c22), -31^o (c37), 00^o (c27), 32^o (c11) and 66^o (c34). As shown in Figure 4.9, the proposed DFL method enables a more accurate pose estimation, and poses estimated through the DFL method are distributed in a continuous region around the true pose.

To demonstrate that a dictionary function can be used for domain estimation for unknown subjects, we use the first 34 subjects in 5 poses c22, c37, c27, c11 and c34 in the PIE dataset for training, and the remaining 34 subjects in the rest 4 poses for testing.



(a) Lighting condition f40 (b) Lighting condition f45 (c) Lighting condition f51

Figure 4.11: Illumination estimation in the Extended YaleB face dataset.

We learn from the training data a dictionary function across pose over the first 34 subjects. As shown in Figure 4.10, the proposed DFL method provides a more accurate continuous pose estimation.

4.3.3.2 Illumination Estimation

In this set of experiments, given a face image in the Extended YaleB dataset, we estimate the azimuth and elevation of the single light source direction. We randomly select 50% (32) of the lighting conditions in the Extended YaleB dataset to learn a dictionary function across illumination over all 34 subjects. The remaining 32 lighting conditions are used for testing. For the SRC method and for each training illumination condition, we form a dictionary from the training data using all 34 subjects. We perform illumination estimation in a similar way as pose estimation. Figure 4.11a, 4.11b, and 4.11c show the illumination estimation for several example lighting conditions. The proposed DFL method provides reasonable estimation to the actual light source directions.

4.4 Conclusion

We presented a general dictionary function learning framework to transform a dictionary learned from one domain to the other. Domain dictionaries are modeled by a parametric function. The dictionary function parameters and domain-invariant sparse codes are then jointly learned by solving an optimization problem with a sparsity constraint. Extensive experiments on real datasets demonstrate the effectiveness of our approach on applications such as pose alignment, pose and illumination estimation and face recognition. The proposed framework can be generalized for non-linearizeable dictionary functions.

Input: a dictionary function $F(\theta, \mathbf{W})$, an image $\mathbf{y}$, domain parameter matrix $\boldsymbol{\Theta}$

Output: an S-dimensional domain parameter vector $\boldsymbol{\theta_y}$ associated with $\mathbf{y}$

begin

1. Initialize with mean domain parameter vector: $\boldsymbol{\theta_y}$ = mean($\boldsymbol{\Theta}$) ;

2. Estimate $\theta^{(s)}$, the s^{th} value in $\boldsymbol{\theta_y}$;

for $s \leftarrow 1$ **to** S **do**

3. Obtain the value range to estimate $\theta^{(s)}$

$\theta^{(s)}_{min} = \min$ (s^{th} row of $\boldsymbol{\Theta}$) ;

$\theta^{(s)}_{max} = \max$ (s^{th} row of $\boldsymbol{\Theta}$) ;

$\theta^{(s)}_{mid} = (\theta^{(s)}_{max} + \theta^{(s)}_{max})/2$;

4. Estimate $\theta^{(s)}$ via a search for the parameters to best represent $\mathbf{y}$.

repeat

$\boldsymbol{\theta_{min}} \leftarrow$ replace the s^{th} value of $\boldsymbol{\theta_y}$ with $\theta^{(s)}_{min}$;

$\boldsymbol{\theta_{max}} \leftarrow$ replace the s^{th} value of $\boldsymbol{\theta_y}$ with $\theta^{(s)}_{max}$;

$\mathbf{x_{min}} \leftarrow \min_{\mathbf{x}} |\mathbf{y} - F(\boldsymbol{\theta_{min}}, \mathbf{W})|_2^2, \;\; s.t. |\mathbf{x}|_o \leq$ t (sparsity) ;

$\mathbf{x_{max}} \leftarrow \min_{\mathbf{x}} |\mathbf{y} - F(\boldsymbol{\theta_{max}}, \mathbf{W})|_2^2, \;\; s.t. |\mathbf{x}|_o \leq$ t (sparsity) ;

$\mathbf{r_{min}} \leftarrow \mathbf{y} - F(\boldsymbol{\theta_{min}}, \mathbf{W})\mathbf{x_{min}}$;

$\mathbf{r_{max}} \leftarrow \mathbf{y} - F(\boldsymbol{\theta_{max}}, \mathbf{W})\mathbf{x_{max}}$;

if $\mathbf{r_{min}} \leq \mathbf{r_{max}}$ **then**

$\theta^{(s)}_{max} = \theta^{(s)}_{mid}$;

else

$\theta^{(s)}_{min} = \theta^{(s)}_{mid}$;

end

$\theta^{(s)}_{mid} = (\theta^{(s)}_{max} + \theta^{(s)}_{max})/2$;

until $|\theta^{(s)}_{max} - \theta^{(s)}_{min}| \leq$ *threshold*;

$\theta^{(s)} \leftarrow \theta^{(s)}_{mid}$;

end

7. return $\boldsymbol{\theta_y}$;

end

Algorithm 5: Domain parameters estimation.

Chapter 5

Compositional Dictionaries for Domain Adaptive Face Recognition

5.1 Introduction

Many image recognition algorithms often fail while experiencing a significant visual domain shift, as they expect the test data to share the same underlying distribution as the training data. A visual domain shift is common and natural in the context of face recognition. Such domain shift is due to changes in poses, illumination, resolution, etc.. Domain adaptation [92] is a promising methodology for handling the domain shift by utilizing knowledge in the source domain for problems in a different but related target domain. [93] is one of the earliest works on semi-supervised domain adaptation, where they model data with three underlying distributions: source domain data distribution, target domain data distribution and a distribution of data that is common to both domains. [94] follows a similar model in handling view point changes in the context of activity recognition, where they assume some activities are observed in both source and target domains, while some other activities are only in one of the domains. Under the above assumption, certain hyperplane-based features trained in the source domain are adapted to the target domain for improved classification. Domain adaptation for object recognition is studied in [95], where the subspaces of the source domain, the target domain and the potential intermediate domains are modeled as points on the Grassmann manifold. The shift between domains is learned by exploiting the geometry of the underlying manifolds. A good survey

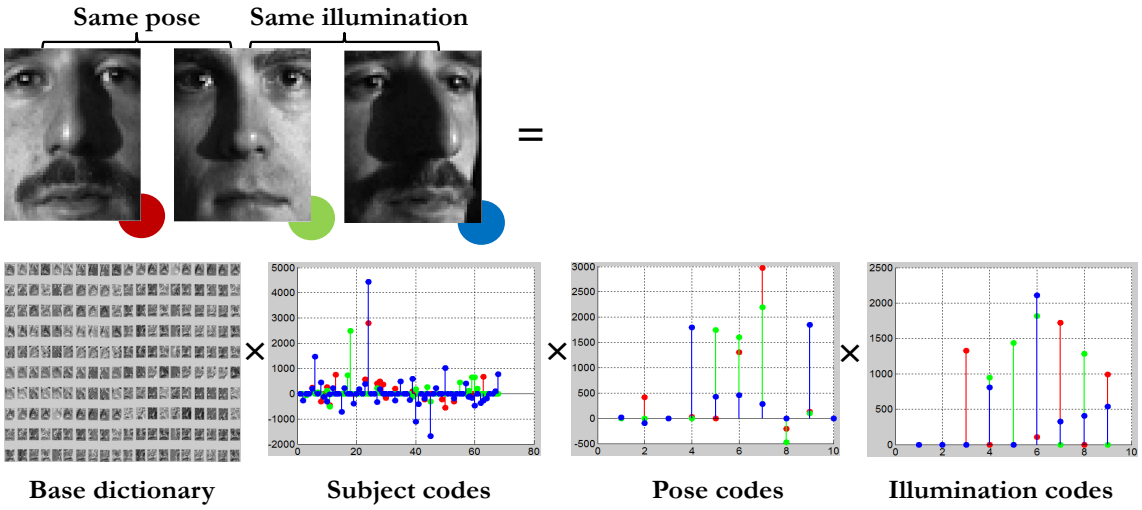


Figure 5.1: Trilinear sparse decomposition. Given a domain base dictionary, an unknown face image is decomposed into sparse representations for each subject, pose and illumination respectively. The domain-invariant subject (sparse) codes are used for pose and illumination insensitive face recognition. The pose and illumination codes are also used to estimate the pose and lighting condition of a given face. Composing subject codes with corresponding domain codes enables pose alignment and illumination normalization.

on domain adaptation can be found in [95].

Face recognition across domain, e.g., pose and illumination, has proved to be a challenging problem [6, 18, 19]. In [18], the eigen light-field (ELF) algorithm is presented for face recognition across pose and illumination. This algorithm operates by estimating the eigen light field or the plenoptic function of the subject's head using all the pixels of various images. In [8, 19], face recognition across pose is performed using stereo matching distance (SMD). The cost to match a probe image to a gallery image is used to evaluate the similarity of the two images. Both ELF and SMD methods are state-of-the-art methods for face recognition across pose and/or illumination variations. Our proposed domain

adaptive dictionary learning approach shows comparable performance to these two methods for face recognition across domain shifts due to pose and illumination variations. In addition, our approach can also be used to estimate the pose and lighting condition of a face, and to perform pose alignment and illumination normalization.

The approach presented here shares some of the attributes of the Tensorfaces method proposed in [6,7,96], but significantly differs in many aspects. In the Tensorfaces method, face images observed in different domains, i.e., faces imaged in different poses under different illuminations, form a face tensor. Then a multilinear analysis is performed on the face tensor using the N-mode SVD decomposition to obtain a core tensor and multiple mode matrices, each for a different domain aspect. The N-mode SVD decomposition is similar to the proposed multilinear sparse decomposition shown in Fig. 5.1, where a given unknown image is decomposed into multiple sparse representations for the given subject, pose and illumination respectively. However, we show through experiments that our method based on sparse decomposition significantly outperforms the N-mode SVD decomposition for face recognition across pose and illumination. Another advantage of the proposed method approach over Tensorfaces is that, the proposed approach provides explicit sparse representations for each subject and each visual domain, which can be used for subject classification and domain estimation. Instead, Tensorfaces performs subject classification through exhaustive projections and matchings. Another work similar to Tensorfaces is discussed in [97], where a bilinear analysis is presented for face matching across domains. In [97], a 2-mode SVD decomposition is first performed and then a Gaussian mixture model is employed to classify subjects. Tensorfaces can be considered as an extension of this work to enable multilinear analysis to face images.

This chapter makes the following contributions:

- We learn a domain base dictionary, and describe each visual domain shift as a sparse representation over the base dictionary.
- We express the dictionary adapted to each domain as sparse linear combinations of the base dictionary.
- We learn for each subject a domain invariant sparse representation.
- We perform pose alignment and illumination normalization by composing sparse representations for subjects and domains.

The remainder of the chapter is organized as follows: Section 5.2 discusses some details about sparse decomposition and multilinear image analysis. In Section 5.3, we formulate the domain adaptive dictionary learning problem for face recognition. In Section 5.4, we present the proposed domain adaptive dictionary learning approach, which consists of algorithms to learn a domain base dictionary, and perform domain invariant sparse coding. Experimental evaluations are given in Section 5.5 on two public face datasets. Finally, Section 5.6 concludes the chapter.

5.2 Background

5.2.1 Sparse Decomposition

Sparse signal representations have recently drawn much attention in vision, signal and image processing [55], [56], [57], [4], [98]. This is mainly due to the fact that signals and images of interest can be sparse in some dictionary. Given an over-complete dictionary

$\mathbf{D}$ and a signal $\mathbf{y}$, finding a sparse representation of $\mathbf{y}$ in $\mathbf{D}$ entails solving the following optimization problem

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ subject to } \mathbf{y} = \mathbf{D}\mathbf{x}, \tag{5.1}$$

where the ℓ_0 sparsity measure $\|\mathbf{x}\|_0$ counts the number of nonzero elements in the vector $\mathbf{x}$. Problem (5.1) is NP-hard and cannot be solved in a polynomial time. Hence, approximate solutions are usually sought [57], [59], [60], [99].

The dictionary $\mathbf{D}$ can be either based on a mathematical model of the data [57] or it can be trained directly from the data [62]. It has been observed that learning a dictionary directly from training rather than using a predetermined dictionary (such as wavelet or Gabor) usually leads to better representation and hence can provide improved results in many practical applications such as restoration and classification [55], [56].

Various algorithms have been developed for the task of training a dictionary from examples. One of the most commonly used algorithms is the K-SVD algorithm [20]. Let $\mathbf{Y}$ be a set of N input signals in a n-dimensional feature space $\mathbf{Y} = [\mathbf{y_1}...\mathbf{y_N}],\ \mathbf{y_i} \in \mathbb{R}^n$. In K-SVD, a dictionary with a fixed number of K items is learned by finding a solution iteratively to the following problem:

$$\arg\min_{\mathbf{D},\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \qquad s.t.\ \forall i, \|\mathbf{x_i}\|_0 \leq T \tag{5.2}$$

where $\mathbf{D} = [\mathbf{d_1}...\mathbf{d_K}],\ \mathbf{d_i} \in \mathbb{R}^n$ is the learned dictionary, $\mathbf{X} = [\mathbf{x_1}, ..., \mathbf{x_N}],\ \mathbf{x_i} \in \mathbb{R}^K$ are the sparse codes of input signals $\mathbf{Y}$, and T specifies the sparsity that each signal has fewer than T items in its decomposition. Each dictionary item $\mathbf{d_i}$ is L_2-normalized.

5.2.2 Multilinear Image Analysis

Linear methods are popular in facial image analysis, such as principal components analysis (PCA) [91], independent component analysis (ICA) [100], and linear discriminant analysis (LDA) [101]. These conventional linear analysis methods work best when variations in domains, such as pose and illumination, are not present. When any visual domain is allowed to vary, the linear subspace representation above does not capture such variation well.

Under the assumption of Lambertian reflectance, Basri and Jacobs [102] have shown that images of an object obtained under a wide variety of lighting conditions can be approximated accurately with a 9-dimensional linear subspace. [103] utilizes the fact that 2D harmonic basis images at different poses are related by close-form linear transformations [104], [105], and extends the 9-dimensional illumination linear space with additional pose information encoded in a linear transformation matrix. The success of these methods suggests the feasibility of decomposing a face image into separate representations for subject and individual domains, e.g. associated pose and illumination, through multilinear algebra.

A multilinear image analysis approach, called Tensorfaces, has been discussed in [6], [7], [96]. Tensor is a multidimensional generalization of a matrix. An N-order tensor $\mathcal{D}$ is an N-dimensional matrix comprising N spaces. N-mode SVD, illustrated in Fig. 5.2, is an extension of SVD that decomposes the tensor as the product of N-orthogonal spaces, where Tensor $\mathcal{Z}$, the core tensor, is analogous to the diagonal singular value matrix in SVD. Mode matrix $\mathbf{U_n}$ contains the orthonormal vectors spanning the column space of

mode-n flattening of $\mathcal{D}$, i.e., the rearranged tensor elements that form a regular matrix [6].

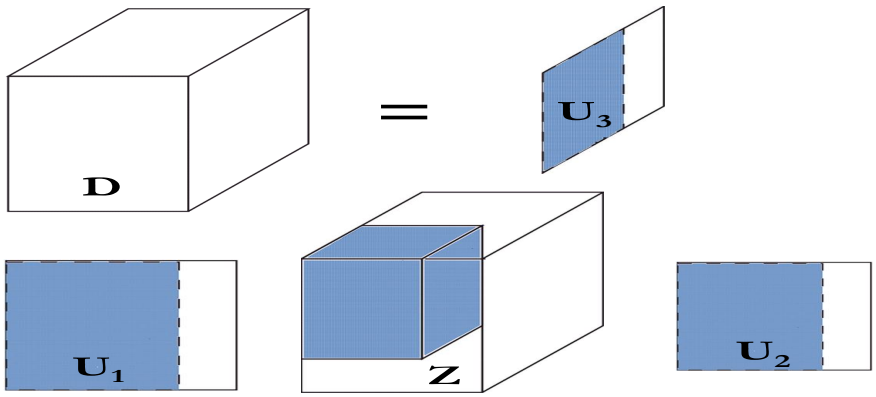


Figure 5.2: An N-mode SVD (N=3 is illustrated) [6].

Consider the the illustration example presented in [6]. Given faces images of 28 subjects, in 5 poses, 3 illuminations and 3 expressions, and each image contains 7943 pixels, we obtain a face tensor $\mathcal{D}$ of size $28 \times 5 \times 3 \times 3 \times 7943$. Suppose we apply a multilinear analysis to the face tensor $\mathcal{D}$ using the 5-mode decomposition as (5.3).

$$\mathcal{D} = \mathcal{Z} \times \mathbf{U}_{\mathbf{subject}} \times \mathbf{U}_{\mathbf{pose}} \times \mathbf{U}_{\mathbf{illum}} \times \mathbf{U}_{\mathbf{expre}} \times \mathbf{U}_{\mathbf{pixels}} \tag{5.3}$$

where the $28 \times 5 \times 3 \times 3 \times 7943$ core tensor $\mathcal{Z}$ governs the interaction between the factors represented in the 5 mode matrices, and each of the mode matrix $\mathbf{U_n}$ represents subjects and respective domains. For example, the k^{th} row of the 28×28 mode matrix $\mathbf{U}_{\mathbf{subject}}$ contains the coefficients for subject k, and the j^{th} row of 5×5 mode matrix $\mathbf{U}_{\mathbf{pose}}$ contains the coefficients for pose j.

Tensorfaces performs subject classification through exhaustive projections and matchings. In the above examples, from the training data, each subject is represented with a 28-sized vector of coefficients to the $28 \times 5 \times 3 \times 3 \times 7943$ base tensor in (5.4)

$$\mathcal{B} = \mathcal{Z} \times \mathbf{U}_{\mathbf{pose}} \times \mathbf{U}_{\mathbf{illum}} \times \mathbf{U}_{\mathbf{expre}} \times \mathbf{U}_{\mathbf{pixels}} \tag{5.4}$$

One can then obtain the basis tensor for a particular pose j, illumination l, and expression e as a $28 \times 1 \times 1 \times 1 \times 7943$ sized subtensor $\mathbf{B_{j,l,e}}$. The subject coefficients of a given unknown face image are obtained by exhaustively projecting this image into a set of candidate basis tensors for every j, l, e combinations. The resulting vector that yields the smallest distance to one of the rows in U_{pose} is adopted as the coefficients for the subject in the test image. In a similar way, one can obtain the coefficient vectors for pose and illumination associated with such test image.

5.3 Problem Formulation

In this section, we formulate the domain adaptive dictionary learning (DADL) approach for face recognition. It is noted that our approach is general and applicable to both image and non-image data. Let $\mathbf{Y}$ denote a set of N signals (face images) in an n-dim feature space $\mathbf{Y} = [\mathbf{y_1}, ..., \mathbf{y_N}],\ \mathbf{y_i} \in \mathbb{R}^n$. Given that face images are from K different subjects $[S_1, \cdots, S_K]$, in J different poses $[P_1, \cdots, P_J]$, and under L different illumination conditions $[I_1, \cdots, I_L]$, $\mathbf{Y}$ can be arranged in six different forms as shown in Fig. 5.3. We assume here that one image is available for each subject under each pose and illumination, i.e., $N = K \times J \times L$.

$\mathbf{A}$ denotes the sparse coefficient matrix of J different poses, $\mathbf{A} = [\mathbf{a_1}, ..., \mathbf{a_J}]$, where $\mathbf{a_j}$ is the sparse representation for the pose P_j. Let $dim(\mathbf{a_j})$ denote the chosen size of sparse code vector $\mathbf{a_j}$, and $dim(\mathbf{a_j}) \leq J$. $\mathbf{B}$ denotes the sparse code matrix of K different subjects, $\mathbf{B} = [\mathbf{b_1}, ..., \mathbf{b_K}]$, where $\mathbf{b_k}$ is the domain invariant sparse representation for the subject S_k, and $dim(\mathbf{b_k}) \leq K$. $\mathbf{C}$ denotes the sparse coefficient matrix of L

different illumination conditions, $\mathbf{C} = [\mathbf{c_1}, ..., \mathbf{c_L}]$, where $\mathbf{c_l}$ is the sparse representation for the illumination condition I_l and $dim(\mathbf{c_l}) \leq L$. The domain base dictionary $\mathbf{D}$ contains $dim(\mathbf{a_j}) \times dim(\mathbf{b_k}) \times dim(\mathbf{c_l})$ atoms arranging in a similar way as Fig. 5.3. Each dictionary atom is in the $\mathbb{R}^n$ space.

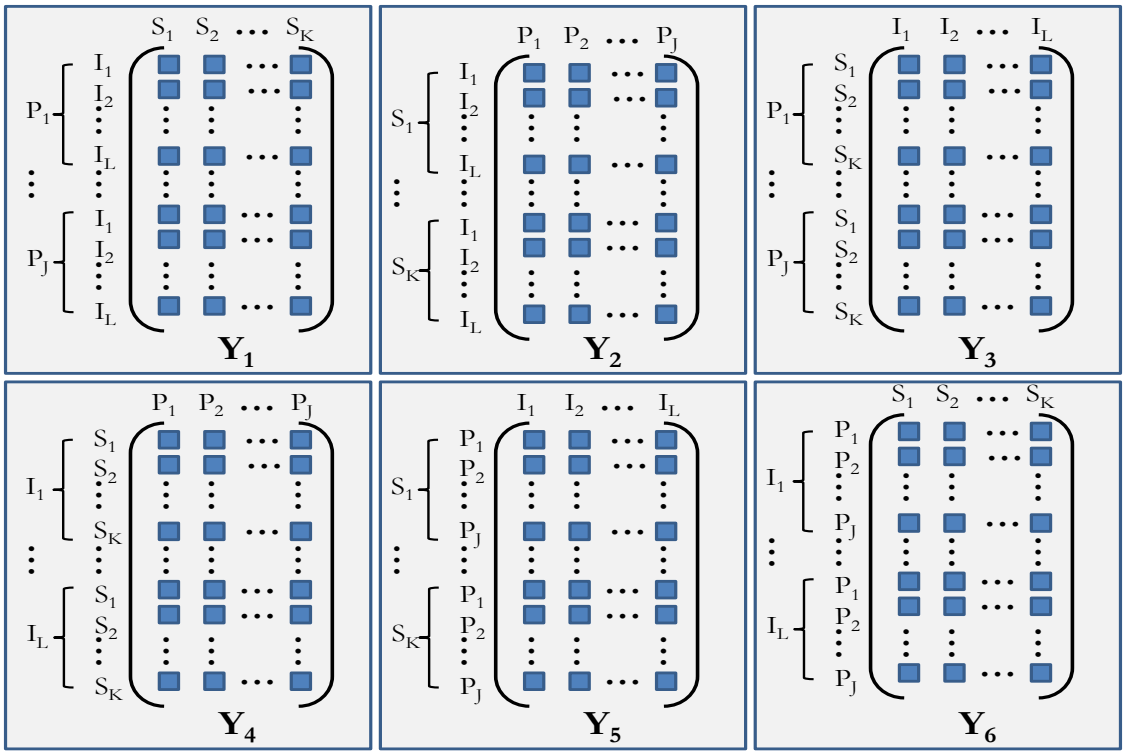


Figure 5.3: Six forms of arranging face images of K subjects in J poses under L illumination conditions. Each square denotes a face image in a column vector form.

Any of the six forms in Fig. 5.3 can be transformed into another through a sequence of vector transpose operations. A vector transpose operation is to consider (stacked) image vectors in Fig. 5.3 as values and perform typical matrix transpose operation. For simplicity, we define six aggregated vector transpose operations $\{T_i\}_{i=1}^{6}$. For example, T_i transforms an input matrix, which is in any of the six forms, into the i-th form defined in Fig. 5.3.

Let $\mathbf{y_k^{jl}}$ be a face image of subject S_k in pose P_j under illumination I_l. The dictionary adapted to pose P_j and illumination I_l is expressed as

$$[[\mathbf{D}^{T_2}\mathbf{a_j}]^{T_3}\mathbf{c_l}]^{T_1}.$$

$\mathbf{y_k^{jl}}$ can be sparsely represented using this dictionary as,

$$\mathbf{y_k^{jl}} = [[\mathbf{D}^{T_2}\mathbf{a_j}]^{T_3}\mathbf{c_l}]^{T_1}\mathbf{b_k},$$

where the subject sparse codes $\mathbf{b_k}$ are independent of both P_j and I_l. In this way, we can represent Fig. 5.3 in a compact matrix form as shown in (5.5).

$$\mathbf{Y_1} = [[\mathbf{D}^{T_3}\mathbf{C_1}]^{T_2}\mathbf{A_1}]^{T_1}\mathbf{B_1} \tag{5.5a}$$

$$\mathbf{Y_2} = [[\mathbf{D}^{T_3}\mathbf{C_2}]^{T_1}\mathbf{B_2}]^{T_2}\mathbf{A_2} \tag{5.5b}$$

$$\mathbf{Y_3} = [[\mathbf{D}^{T_1}\mathbf{B_3}]^{T_2}\mathbf{A_3}]^{T_3}\mathbf{C_3} \tag{5.5c}$$

$$\mathbf{Y_4} = [[\mathbf{D}^{T_1}\mathbf{B_4}]^{T_3}\mathbf{C_4}]^{T_2}\mathbf{A_4} \tag{5.5d}$$

$$\mathbf{Y_5} = [[\mathbf{D}^{T_2}\mathbf{A_5}]^{T_1}\mathbf{B_5}]^{T_3}\mathbf{C_5} \tag{5.5e}$$

$$\mathbf{Y_6} = [[\mathbf{D}^{T_2}\mathbf{A_6}]^{T_3}\mathbf{C_6}]^{T_1}\mathbf{B_6} \tag{5.5f}$$

The proposed domain adaptive dictionary model is built as follows,

- We learn a base dictionary $\mathbf{D}$ that is independent of subjects and domains.
- We learn a sparse representation over the base dictionary for each visual domain, e.g., a specific pose or illumination condition.
- We express the dictionary adapted to a specific domain as sparse linear combinations of the base dictionary using sparse representation of the domain under consideration..

- We learn for each subject a domain invariant sparse representation.

We now provide the details of solutions to the following two problems

- How to learn a base dictionary that is independent of subject and domains.
- Given an input face image and the base dictionary, how to obtain the sparse representation for the associated pose and illumination, and the domain invariant sparse representation for the subject.

5.4 Domain Adaptive Dictionary Learning

In this section, we first show, given a domain base dictionary $\mathbf{D}$, sparse coefficient matrices $\{\mathbf{A_i}\}_{i=1}^{6}$, $\{\mathbf{B_i}\}_{i=1}^{6}$ and $\{\mathbf{C_i}\}_{i=1}^{6}$ are equivalent across different equations in (5.5). Then, we present algorithms to learn a domain base dictionary $\mathbf{D}$, and perform domain invariant sparse coding.

5.4.1 Equivalence of Six Forms

To learn a domain base dictionary $\mathbf{D}$, we first need to establish the following proposition.

Proposition: Given a domain base dictionary $\mathbf{D}$, matrices $\{\mathbf{A_i}\}_{i=1}^{6}$ in all six equations in (5.5) are equivalent, and so are matrices $\{\mathbf{B_i}\}_{i=1}^{6}$ and $\{\mathbf{C_i}\}_{i=1}^{6}$.

First we show matrices $\mathbf{B_i}$ in (5.5a) and (5.5f) are equivalent. $\mathbf{Y_1}$ and $\mathbf{Y_6}$ in Fig. 5.3 are different only in the row order. We assume a permutation matrix $\mathbf{P_{16}}$ will permutate the rows of $\mathbf{Y_1}$ into $\mathbf{Y_6}$, i.e., $\mathbf{P_{16}Y_1} = \mathbf{Y_6}$. Through a dictionary learning process, e.g., k-SVD [20], we obtain a dictionary $\mathbf{D_1}$ and the associated sparse code matrix $\mathbf{B_1}$ for $\mathbf{Y_1}$.

$\mathbf{Y_1}$ can be reconstructed as $\mathbf{Y_1} = \mathbf{D_1 B_1}$. We change the row order of $\mathbf{D_1}$ according to $\mathbf{P_{16}}$ without modifying the actual atom value as $\mathbf{D_6} = \mathbf{P_{16} D_1}$. We decompose $\mathbf{Y_6}$ using $\mathbf{D_6}$ as $\mathbf{Y_6} = \mathbf{D_6 B_6}$, i.e., $\mathbf{P_{16} Y_1} = \mathbf{P_{16} D_1 B_6}$, and we have $\mathbf{B_1} = \mathbf{B_6}$.

Then we show that matrices $\mathbf{A_i}$, $\mathbf{B_i}$ and $\mathbf{C_i}$ in (5.5a) and (5.5b) are equivalent. If we stack all the images from the same subject under the same pose but different illumination as a single observation, we can consider $\mathbf{Y_2} = \mathbf{Y_1}^T$. By assuming a bilinear model, we can represent $\mathbf{Y_1}$ as $\mathbf{Y_1} = [\mathbf{D_c A_1}]^T \mathbf{B_1}$, and we have $\mathbf{Y_2} = \mathbf{Y_1}^T = [\mathbf{D_c}^T \mathbf{B_1}]^T \mathbf{A_1}$. As $\mathbf{Y_2} = [\mathbf{D_c}^T \mathbf{B_2}]^T \mathbf{A_2}$, $\mathbf{A_i}$ and $\mathbf{B_i}$ are equivalent in (5.5a) and (5.5b). As both equations share a bilinear map $\mathbf{D}^{T_3} \mathbf{C_i}$, with a common base dictionary $\mathbf{D}$, matrices $\mathbf{C_i}$ are also equivalent in (5.5a) and (5.5b).

Finally, we show matrices $\mathbf{A_i}$ and $\mathbf{C_i}$ in (5.5a) and (5.5f) are equivalent. We have shown in (5.5a) and (5.5f) that matrices $\mathbf{B_i}$ are equivalent. $[[\mathbf{D}^{T_3} \mathbf{C_1}]^{T_2} \mathbf{A_1}]^{T_1}$ and $[[\mathbf{D}^{T_2} \mathbf{A_6}]^{T_3} \mathbf{C_6}]^{T_1}$ are different only in the row order. We can use the bilinear model argument made above to easily show that matrices $\mathbf{A_i}$ and $\mathbf{C_i}$ are equivalent in (5.5a) and (5.5f).

Through the transitivity of equivalence, we can further show matrices $\mathbf{A_i}$ in all six equations in (5.5) are equivalent, and so are matrices $\mathbf{B_i}$ and $\mathbf{C_i}$. We drop the subscripts in subsequent discussions and denote them as $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$.

5.4.2 Domain Invariant Sparse Coding

As matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are equivalent across all six forms in (5.5) , we propose to learn the base dictionary $\mathbf{D}$ using Algorithm 6 given below. Algorithm 1 is designed as an

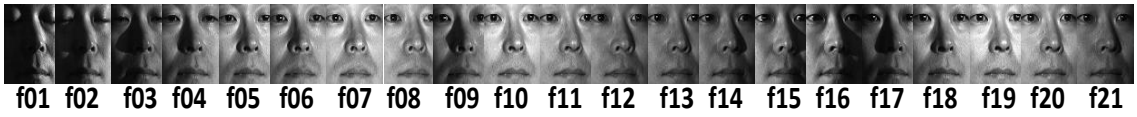
iterative method, and each iteration consists of several typical sparse dictionary learning problems. Thus, this algorithm is flexible and can rely on any sparse dictionary learning methods. We adopt the highly efficient dictionary learning method, k-SVD [20]. It is noted that we can easily omit one domain aspect through dictionary "marginalization". For example, after learning the based dictionary $\mathbf{D}$, we can marginalize over illumination sparse codes matrix $\mathbf{C}$ and adopt $[\mathbf{D}^{T_3}\mathbf{C}]^{T_2}$ as the base dictionary for pose domains only.

With the learned base dictionary $\mathbf{D}$, we can perform domain invariant sparse coding as shown in Algorithm 7. This algorithm accepts any pursuit algorithms, such as OMP [60, 99]. Through this algorithm, an input face image can be decomposed into sparse representations for the associated pose and illumination, and a domain invariant sparse representation for the subject.

Convergence of Algorithms 6 and 7 can be established using the convergence results of k-SVD discussed in [20]. The convergence of both algorithms depends on the success of pursuit algorithms involved in each iteration step. We have observed empirical convergence for both Algorithm 6 and 7 in all the experiments reported below.

5.5 Experimental Evaluation

This section presents experimental evaluations on two public face datasets: the CMU PIE dataset [89] and the Extended YaleB dataset [90]. The PIE dataset consists of 68 subjects imaged simultaneously under 13 different poses and 21 lighting conditions, as shown in Fig. 5.4. The Extended YaleB dataset contains 38 subjects with near frontal pose under 64 lighting conditions. 64×48 sized images are used in the domain composition



(a) Illumination variation

(b) Pose variation

Figure 5.4: Pose and illumination variation in the PIE dataset.

experiments in Section 5.5.2 for clearer visualization. In the remaining experiments, all the face images are resized to 32×24. The proposed Domain Adaptive Dictionary learning method is refereed to as DADL in subsequent discussions.

5.5.1 Learned Domain Base Dictionaries

In our experiments, four different domain base dictionaries $\mathbf{D_{10}}$, $\mathbf{D_4}$, $\mathbf{D_{34}}$, and $\mathbf{D_{32}}$ are learned. We explain here the configurations for each base dictionary.

- $\mathbf{D_4}$: This dictionary is learned from the PIE dataset by using 68 subjects in 4 poses under 21 illumination conditions. The four training poses to the dictionary are $c02$, $c07$, $c09$ and $c14$ poses shown in Fig. 5.4. The coefficient vector sizes for subject, pose and illumination are 68, 4 and 9. The respective coefficient sparsity values, i.e., the maximal number of non-zero coefficients, are 20, 4 and 9.
- $\mathbf{D_{10}}$: This dictionary is learned from the PIE dataset by using 68 subjects in 10

poses under all illumination conditions. The three unknown poses to the dictionary are $c27$ (frontal), $c05$ (side) and $c22$ (profile) poses. The coefficient vector sizes for subject, pose and illumination are 68, 10 and 9. The respective coefficient sparsity values are 20, 8 and 9.

- $\mathbf{D_{34}}$: This dictionary is learned from the PIE dataset by using the first 34 subjects in 13 poses under 21 illumination conditions. The coefficient vector sizes for subject, pose and illumination are 34, 13 and 9. The respective coefficient sparsity values are 12, 8 and 9.
- $\mathbf{D_{32}}$: This dictionary is learned from the Extended YaleB dataset by using 38 subjects under 32 randomly selected lighting conditions. The coefficient vector sizes for subject and illumination are 38, and 32. The respective coefficient sparsity values are 20 and 20.

5.5.2 Domain Composition

Using the proposed trilinear sparse decomposition over a base dictionary as illustrated in Algorithm 7, we extract from a face image the respective sparse representations for subject, pose and illumination. We can translate a subject to a different pose and illumination by composing the corresponding subject and domain sparse codes over the base dictionary. As discussed in Sec. 5.2.2, Tensorfaces also enables the decomposition of a face image into separate coefficients for the subject, pose and illumination through exhaustive projections and matchings. We adopt the Tensorfaces method here for a fair comparison in our domain composition experiments.

5.5.2.1 Pose Alignment

In Fig. 5.5a, the base dictionary $\mathbf{D_{34}}$ is used in the DADL experiments. To enable a fair comparison, we adopt the same training data and sparsity values for $\mathbf{D_{34}}$ in the corresponding Tensorfaces experiments. Given faces from subject s01 under different poses, where both the subject and poses are present in the training data, we extract the subject (sparse) codes for $s01$ from each of them. Then we extract the pose codes for $c27$ (frontal) and the illumination codes for $f05$ from an image of subject $s43$. It is noted that, for such *known subject* cases, the composition $(s01, c27, f05)$ through both DADL and Tensorfaces provides good reconstructions to the ground truth image. The reconstruction using DADL is clearer than the one using Tensorfaces.

In Fig. 5.5b, we first extract the subject codes for $s43$, which is an unknown subject to $\mathbf{D_{34}}$. Then we extract the pose codes and the illumination codes from the set of images of $s01$ in Fig. 5.5a. In this *unknown subject* case, the composition using our DADL method provides significantly more accurate reconstruction to the groudtruth images than the Tensorfaces method. The central assumption in the literature on sparse representation for faces is that the test face image should be represented in terms of training images of the same subject [22], [106]. As $s43$ is unknown to $\mathbf{D_{34}}$, therefore, it is expected that the reconstruction of the subject information is through a linear combination of other known subjects, which is an approximation but not exact.

In Fig. 5.5c, the base dictionary $\mathbf{D_{10}}$ is used in the DADL experiments, and the same training data and sparsity values for $\mathbf{D_{10}}$ are used in the corresponding Tensorfaces experiments. We first extract the subject codes for $s43$. Then we extract the pose codes

for pose $c22$, $c05$ and $c27$, which are unknown poses to the training data. Through domain composition, for such *unknown pose* cases, we obtain more acceptable reconstruction to the actual images using DADL than Tensorfaces. This indicates that, using the proposed DADL method, an unknown pose can be much better approximated in terms of a set of observed poses.

5.5.2.2 Illumination Normalization

In Fig. 5.6a, we use frontal faces from subject $s28$, which is known to $\mathbf{D_{34}}$, under different illumination conditions. For each image, we first isolate the codes for subject, pose and illumination, and then replace the illumination codes with the one for $f11$. If $f11$ is observed in the training data, the illumination codes for $f11$ can be obtained during training. Otherwise, the illumination codes for f11 can be extracted from any face image under $f11$ illumination. It is shown in Fig. 5.6a that, for such *known subject* cases, after removing the illumination variation, we can obtain a reconstructed image close to the ground truth image using both DADL and Tensorfaces.

Subject $s43$ in Fig. 5.6b is unknown to $\mathbf{D_{34}}$. The composed images from DADL exhibit significantly more accurate subject, pose and illumination reconstruction than Tensorfaces. As discussed before, the reconstruction to the subject here is only an approximation but not exact.

5.5.3 Pose and Illumination Invariant Face Recognition

5.5.3.1 Classifying PIE 68 Faces using $\mathbf{D_4}$ and $\mathbf{D_{10}}$

Fig. 5.7 shows the face recognition performance under combined pose and illumination variation for the CMU PIE dataset. To enable the comparison with [8], we adopt the same challenging setup as described in [8]. In this experiment, we classify 68 subjects in three poses, frontal ($c27$), side ($c05$), and profile ($c22$), under all 21 lighting conditions. We select one of the 3 poses as the gallery pose, and one of the remaining 2 poses as the probe pose, for a total of 6 gallery-probe pose pairs. For each pose pair, the gallery is under the lighting condition $f11$ as specified in [8], and the probe is under the illumination indicated in the table. Methods compared here include Tensorface [6, 7], SMD [8], and our method DADL. DADL-4 uses the dictionary $\mathbf{D_4}$ and DADL-10 uses $\mathbf{D_{10}}$. In both DADL-4 and DADL-10 setups, three testing poses $c27$, $c05$, and $c22$ are unknown to the training data. It is noted that, to the best of our knowledge, SMD reports the best recognition performance in such experimental setup. As shown in Fig. 5.7, among 4 out of 6 Gallery-Probe pose pairs, the proposed DADL-10 is better or comparable to SMD.

SMD methods perform classification based on the stereo matching distance between each pair of gallery-probe images. The stereo matching distance becomes more robust when the pose variation between such image pair decreases. However, the proposed DADL classifies faces based on subject codes extracted from each image alone. The robustness of the extracted subject codes only depends on the capability of the base dictionary to reconstruct such a face. This explains why our DADL method significantly outperforms SMD for more challenging pose pairs, e.g., *Profile-Frontal* pair with 62^o

pose variation; but performs worse than SMD for easier pairs, e.g., *Frontal-Side* with 16^o pose variation.

It can be observed in Fig. 5.5c that an unknown pose can be approximated in terms of a set of observed poses. By representing three testing poses through four training poses in $\mathbf{D_4}$, instead of ten poses in $\mathbf{D_{10}}$, we obtain reasonable performance degradations but with 60% less training data.

Though the Tensorface method shares a similar multilinear framework to DADL, as seen from Fig. 5.7, it only handles limited pose and illumination variations.

5.5.3.2 Classifying Extended YaleB using $\mathbf{D_{32}}$

We adopt a similar protocol as described in [26]. In the Extended YaleB dataset, each of the 38 subjects is imaged under 64 lighting conditions. We split the dataset into two halves by randomly selecting 32 lighting conditions as training, and the other half for testing. Fig. 5.8 shows the illumination variation in the testing data. When we learn $\mathbf{D_{32}}$ using Algorithm 6, we also obtain the sparse codes for each subject. During testing, we extract the subject codes from each testing face image and classify it based on the best match in subject codes learned from the training data. As shown in Table 5.1, the proposed DADL method outperforms other state-of-the-art sparse representation methods (The results for other compared methods are taken from [26]). When the extreme illumination conditions are included, we obtain an average recognition rate 98.67%. By excluding two extreme illumination condition $f34$ and $f35$, we obtain an average recognition rate 99.7%.

5.5.4 Pose and Illumination Estimation

In Section 5.5.3, we report the results of experiments over subject codes using base dictionaries $\mathbf{D_{10}}$ and $\mathbf{D_4}$. While generating subject codes, we simultaneously obtain pose codes and illumination codes. Such pose and illumination codes can be used for pose and illumination estimation. In Fig. 5.9, we show the pose and illumination estimation performance on the PIE dataset using the pose and illumination sparse codes through both DADL and Tensorfaces. The proposed DADL method exhibits significantly better domain estimation accuracy than the Tensorfaces method. By examining Fig. 5.9, it can be noticed that the most confusing illumination pairs in DADL, e.g., $(f05, f18)$, $(f10, f19)$ and $(f11, f20)$ are very visually similar based on Fig. 5.4.

5.5.5 Mean Code and Error Analysis

As discussed in Sec. 5.2.2, the Tensorface method shares a similar multilinear framework to the proposed DADL method. However, we showed through the above experiments that the proposed method based on sparse decomposition significantly outperforms the N-mode SVD decomposition for face recognition across pose and illumination. In this section, we analyze in more detail the behaviors of the proposed DADL and Tensorfaces, by comparing subject and domain codes extracted from a face image using these two methods.

For the experiments in this section, we adopt the base dictionary $\mathbf{D_{10}}$ for DADL, and the same training data and sparsity values of $\mathbf{D_{10}}$ for Tensorfaces to learn the core tensor and the associated mode matrices. The same testing data is used for both methods,

i.e., 68 subjects in the PIE dataset under 21 illumination conditions in the $c27$ (frontal), $c05$ (side) and $c22$ (profile) poses, which are three unseen poses not present in the training data.

Fig. 5.10 and Fig. 5.11 shows the mean subject codes of subject $s1$ and $s2$ over 21 illumination conditions in each of the three testing poses, and the associated standard errors. In each of the two figures, we compare the first row, the subject codes from DADL, with the second row, the subject codes from Tensorfaces. We can easily notice the following: first, the subject codes extracted using DADL are more sparse; second, DADL subject codes are more consistent across pose; third, DADL subject codes are more consistent across illumination, which is indicated by the smaller standard errors. By comparing Fig. 5.10 with Fig. 5.11, we also observe that the DADL subject codes are more discriminative. Therefore, face recognition using DADL subject codes significantly outperforms recognition using Tensorfaces subject codes.

Fig. 5.12 shows the mean illumination code of illumination condition $f1$ over 68 subjects in each of the three testing poses, and the associated errors. By comparing the first row with the second row in Fig. 5.12, we find that illumination codes extracted using DADL are more consistent across subject and pose than codes from Tensorfaces. Fig. 5.13 shows the mean pose code of subject $s1$ over 21 illumination conditions for each of the three testing poses, and the associated error. By comparing the first row with the second row in Fig. 5.13, we notice that pose codes from DADL are significantly more consistent across different illumination conditions, indicated by the smaller standard errors.

5.6 Conclusion

We presented an approach to learn domain adaptive dictionaries for face recognition across pose and illumination domain shifts. With a learned domain base dictionary, an unknown face image is decomposed into subject codes, pose codes and illumination codes. Subject codes are consistent across domains, and enable pose and illumination insensitive face recognition. Pose and illumination codes can be used to estimate the pose and lighting condition of the face. The proposed method can be generalized for multilinear face image analysis, however, more experimental validations are needed. We also plan to evaluate the usefulness of our domain adaptive dictionary learning framework in applications other than face recognition.

Input: signals $\mathbf{Y}$, sparsity level T_a, T_b, T_c

Output: domain base dictionary $\mathbf{D}$

begin

Initialization stage:

1. Initialize $\mathbf{B}$ by solving (5.5a) via k-SVD

$\min_{\mathbf{D_b},\mathbf{B}} \|\mathbf{Y_1} - \mathbf{D_b}\mathbf{B}\|_F^2 \; s.t. \; \forall k \; \|\mathbf{b}_k\|_o \leq T_b$, where $\mathbf{D_b} = [[\mathbf{D}^{T_3}\mathbf{C}]^{T_2}\mathbf{A}]^{T_1}$

repeat

2. apply $\mathbf{B}$ to (5.5a) and solve via k-SVD ($\mathbf{B}^\dagger = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T$)

$\min_{\mathbf{D_a},\mathbf{A}} \|(\mathbf{Y_1}\mathbf{B}^\dagger)^{T_2} - \mathbf{D_a}\mathbf{A}\|_F^2 \; s.t. \; \forall j \; \|\mathbf{a}_j\|_o \leq T_A$, where $\mathbf{D_a} = [\mathbf{D}^{T_3}\mathbf{C}]^{T_2}$

3. apply $\mathbf{A}$ to (5.5d) and solve via k-SVD

$\min_{\mathbf{D_c},\mathbf{C}} \|(\mathbf{Y_4}\mathbf{A}^\dagger)^{T_3} - \mathbf{D_c}\mathbf{C}\|_F^2 \; s.t. \; \forall l \; \|\mathbf{c}_l\|_o \leq T_c$, where $\mathbf{D_c} = [\mathbf{D}^{T_1}\mathbf{B}]^{T_3}$

4. apply $\mathbf{C}$ to (5.5e) and solve via k-SVD

$\min_{\mathbf{D_b},\mathbf{B}} \|(\mathbf{Y_5}\mathbf{C}^\dagger)^{T_1} - \mathbf{D_b}\mathbf{B}\|_F^2 \; s.t. \; \forall k \; \|\mathbf{b}_k\|_o \leq T_b$, where $\mathbf{D_b} = [\mathbf{D}^{T_2}\mathbf{A}]^{T_1}$

until *convergence*;

5. Design the domain base dictionary:

$\mathbf{D} \leftarrow [\mathbf{D}^{T_2}\mathbf{A}]\mathbf{A}^\dagger$;

6. return $\mathbf{D}$;

end

Algorithm 6: Domain base dictionary learning.

Input: an input image $\mathbf{y}$, domain base dictionary $\mathbf{D}$, sparsity level T_a, T_b, T_c

Output: sparse representation vector for pose $\mathbf{a}$, illumination $\mathbf{c}$, subject $\mathbf{b}$

begin

Initialization stage:

1. Initialize domain sparse code vector $\mathbf{a}$ and $\mathbf{c}$ with random values;

Sparse coding stage:

repeat

2. apply $\mathbf{a}$ and $\mathbf{c}$ to (5.5a) and obtain $\mathbf{b}$ via any pursuit algorithm,

$$\min_{\mathbf{b}} \|\mathbf{y} - [[\mathbf{D}^{T_3}\mathbf{c}]^{T_2}\mathbf{a}]^T\mathbf{b}\|_2^2 \; s.t. \; \|\mathbf{b}\|_o \leq T_b,$$

3. apply $\mathbf{b}$ and $\mathbf{c}$ to (5.5d) and obtain $\mathbf{a}$ via any pursuit algorithm,

$$\min_{\mathbf{a}} \|\mathbf{y} - [[\mathbf{D}^{T_1}\mathbf{b}]^{T_3}\mathbf{c}]^T\mathbf{a}\|_2^2 \; s.t. \; \|\mathbf{a}\|_o \leq T_a,$$

4. apply $\mathbf{a}$ and $\mathbf{b}$ to (5.5e) and obtain $\mathbf{c}$ via any pursuit algorithm,

$$\min_{\mathbf{c}} \|\mathbf{y} - [[\mathbf{D}^{T_2}\mathbf{a}]^{T_1}\mathbf{b}]^T\mathbf{c}\|_2^2 \; s.t. \; \|\mathbf{c}\|_o \leq T_c,$$

until *convergence*;

5. return

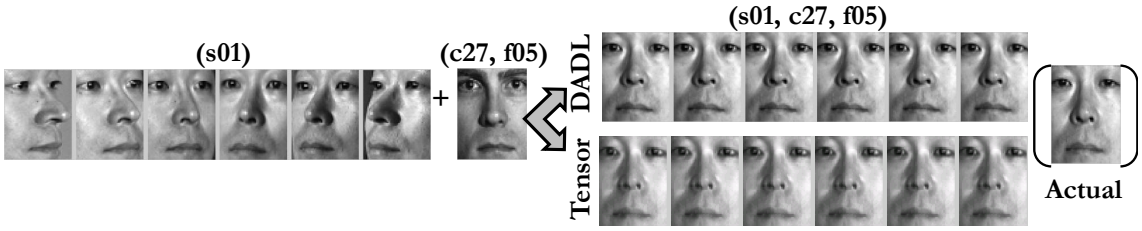
domain invariant sparse codes for the input subject: $\mathbf{b}$,

sparse codes for the input pose: $\mathbf{a}$,

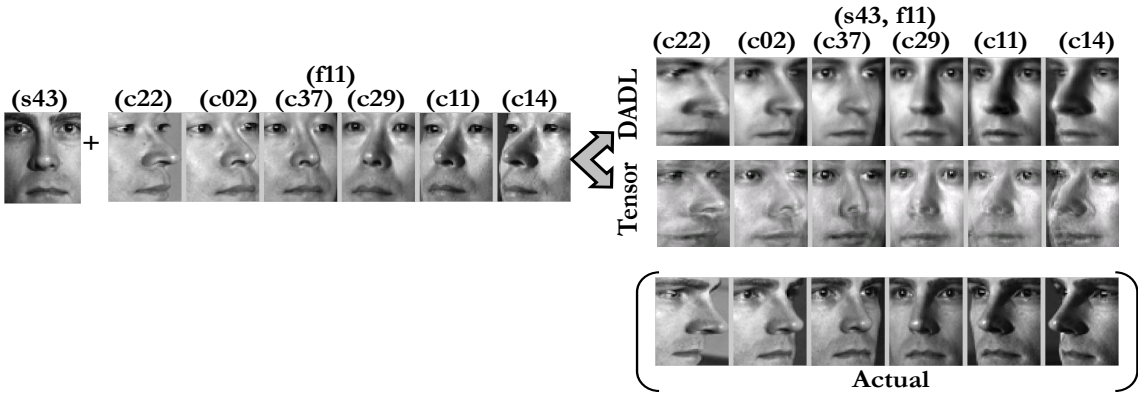
sparse codes for the input illumination: $\mathbf{c}$;

end

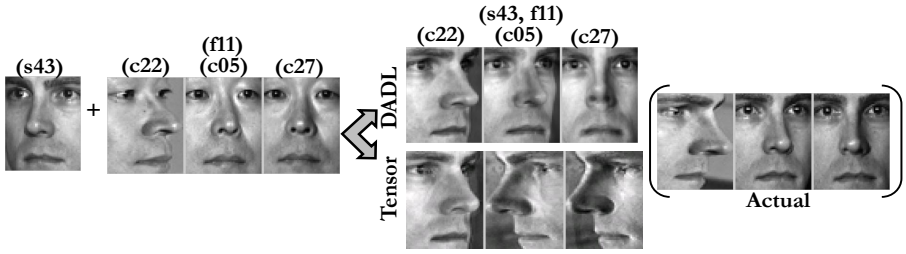
Algorithm 7: Domain invariant sparse coding for a face image.



(a) Composition using base dictionary $\mathbf{D_{34}}$. s01 is a known subject to $\mathbf{D_{34}}$.

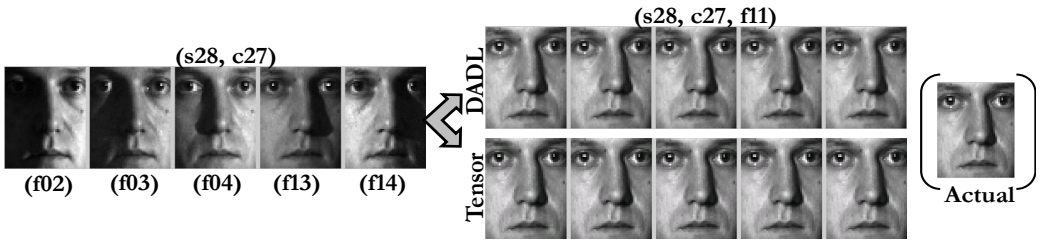


(b) Composition using base dictionary $\mathbf{D_{34}}$. s43 is an unknown subject to $\mathbf{D_{34}}$.

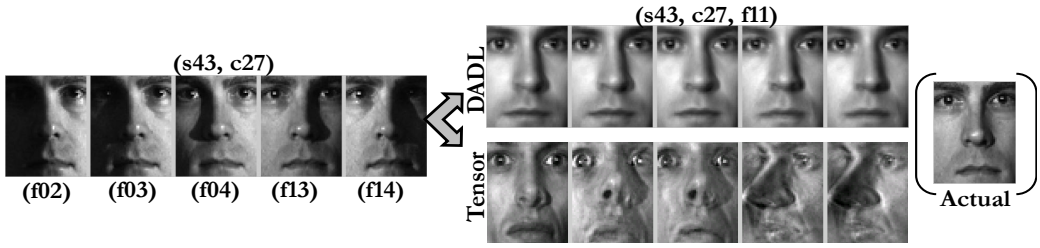


(c) Composition using base dictionary $\mathbf{D_{10}}$. c22, c05 and c27 are unknown poses to $\mathbf{D_{10}}$.

Figure 5.5: Pose alignment through domain composition. In each corresponding Tensorfaces experiment, we adopt the same training data and sparsity values used for the DADL base dictionary for a fair comparison. When a subject or a pose is unknown to the training data, the proposed DADL method provides significantly more accurate reconstruction to the ground truth images.



(a) Composition using base dictionary $\mathbf{D_{34}}$. s28 is a known subject to $\mathbf{D_{34}}$.

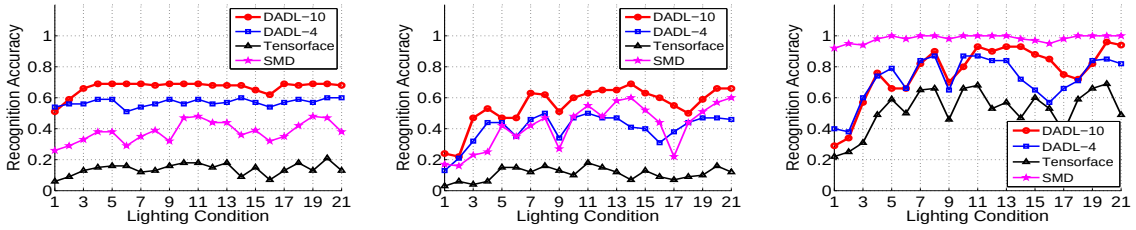


(b) Composition using base dictionary $\mathbf{D_{34}}$. s43 is an unknown subject to $\mathbf{D_{34}}$.

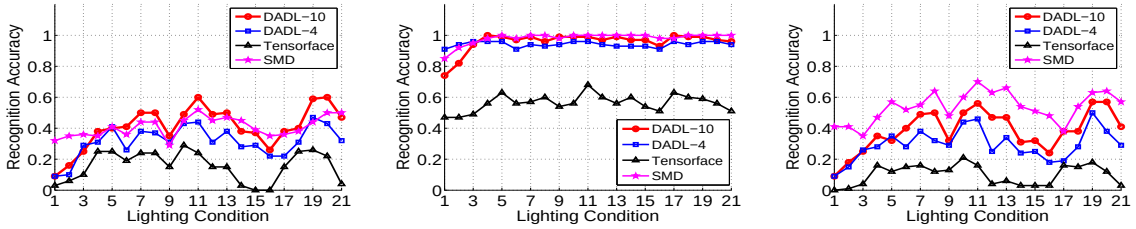
Figure 5.6: Illumination normalization through domain composition. In each corresponding Tensorfaces experiment, we adopt the same training data and sparsity values used for the DADL base dictionary for a fair comparison. When a subject is unknown to the training data, the proposed DADL method provides significantly more accurate reconstruction to the ground truth images.

Table 5.1: Face recognition rate (%) on the Extended YaleB face dataset across 32 different lighting conditions. By excluding two extreme illumination condition f34 and f35, we obtain an average recognition rate 99.7%

DADL	D-KSVD [24]	LC-KSVD [26]	K-SVD [20]	SRC [22]	LLC [80]
98.67	94.10	95.00	93.1	80.5	90.7



(a) Gallery: profile. Probe: frontal. (b) Gallery: profile. Probe: side. (c) Gallery: frontal. Probe: side.



(d) Gallery: frontal. Probe: profile. (e) Gallery: side. Probe: frontal. (f) Gallery: side. Probe: profile.

Figure 5.7: Face recognition under combined pose and illumination variations for the CMU PIE dataset. Given three testing poses, Frontal ($c27$), Side ($c05$), Profile ($c22$), we show the percentage of correct recognition for each disjoint pair of Gallery-Probe poses. See Fig. 5.4 for poses and lighting conditions. Methods compared here include Tensorface [6, 7], SMD [8] and our domain adaptive dictionary learning (DADL) method . DADL-4 uses the dictionary $\mathbf{D_4}$ and DADL-10 uses $\mathbf{D_{10}}$. To the best of our knowledge, SMD reports the best recognition performance in such experimental setup. 4 out of 6 Gallery-Probe pose pairs, i.e., (a), (b), (d) and (e), our results are comparable to SMD.

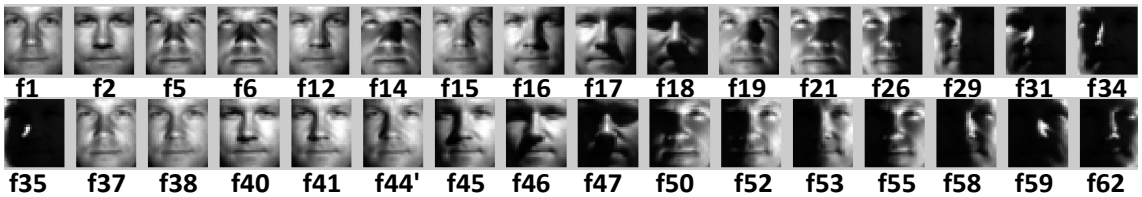
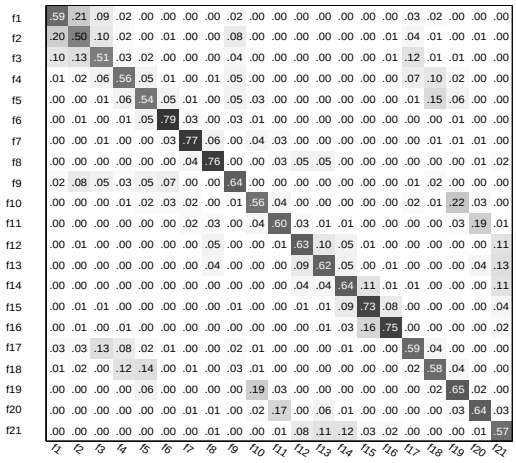
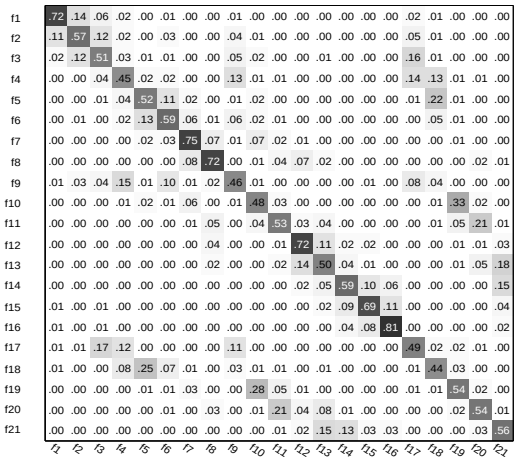


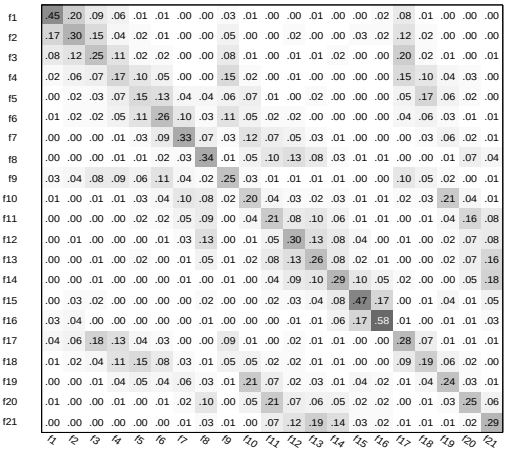
Figure 5.8: Illumination variation in the Extended YaleB dataset.



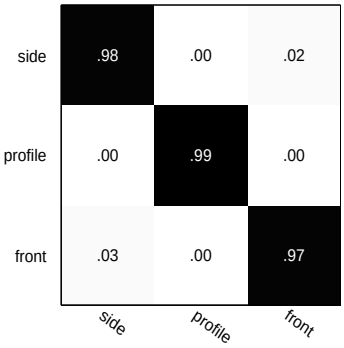
(a) Illumination estimation with $\mathbf{D_{10}}$



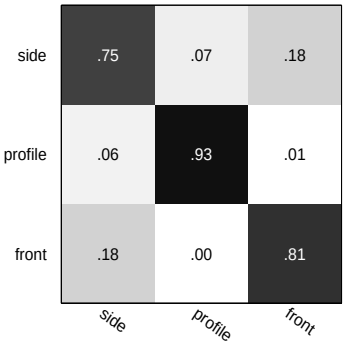
(b) Illumination estimation with $\mathbf{D_4}$



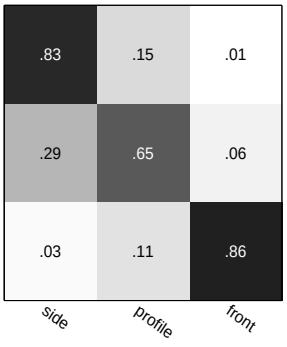
(c) Illumination estimation with Tensorfaces



(d) Pose estimation with $\mathbf{D_{10}}$

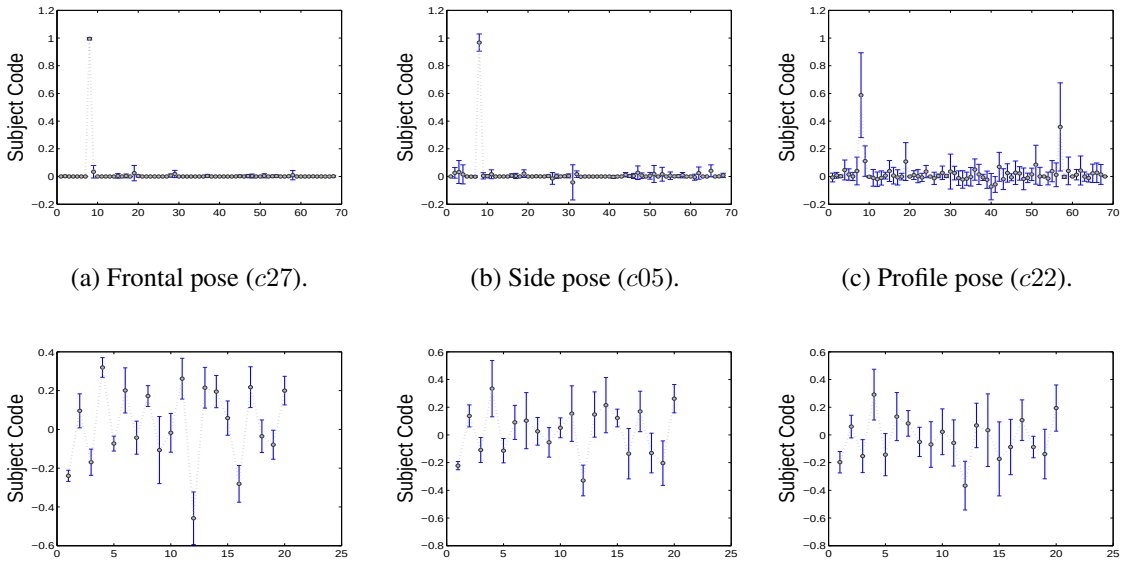


(e) Pose estimation with $\mathbf{D_4}$



(f) Pose estimation with Tensorfaces

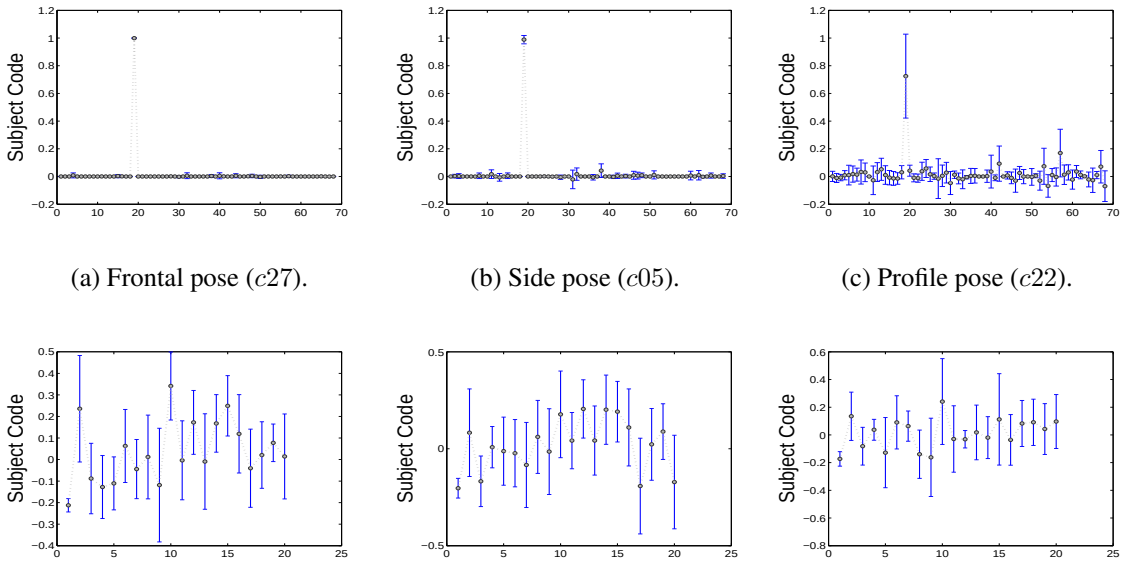
Figure 5.9: Illumination and pose estimation on the CMU PIE dataset using base dictionaries $\mathbf{D_4}$ and $\mathbf{D_{10}}$. Average accuracy: (a) 0.63, (b) 0.58, (c) 0.28, (d) 0.98, (e) 0.83, (f) 0.78. The proposed DADL method exhibits significantly better domain estimation accuracy than the Tensorfaces method.



(a) Frontal pose ($c27$). (b) Side pose ($c05$). (c) Profile pose ($c22$).

(d) Frontal pose ($c27$). (e) Side pose ($c05$). (f) Profile pose ($c22$).

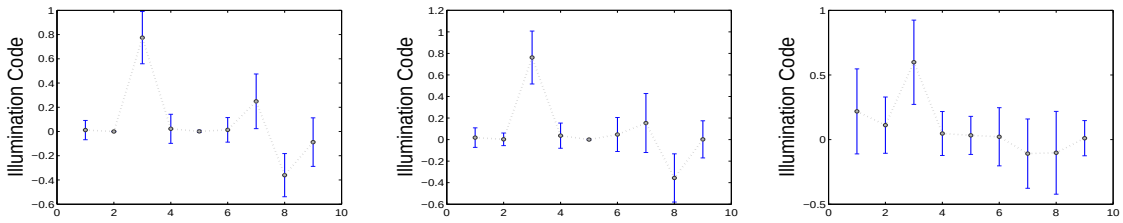
Figure 5.10: Mean subject code of subject $s1$ over 21 illumination conditions in each of the three testing poses, and standard error of the mean code. (a),(b),(c) are generated using DADL with the base dictionary $\mathbf{D_{10}}$. (d),(e),(f) are generated using Tensorfaces.



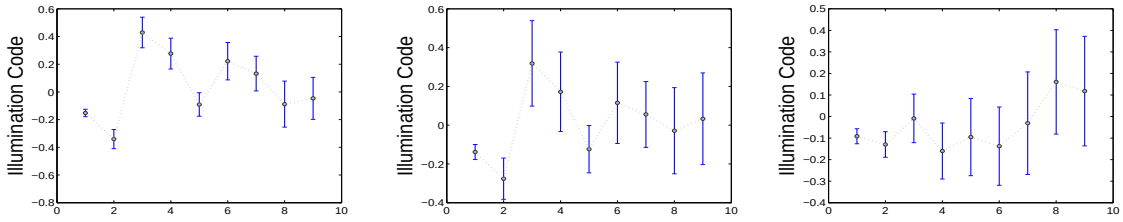
(a) Frontal pose ($c27$). (b) Side pose ($c05$). (c) Profile pose ($c22$).

(d) Frontal pose ($c27$). (e) Side pose ($c05$). (f) Profile pose ($c22$).

Figure 5.11: Mean subject code of subject $s2$ over 21 illumination conditions in each of the three testing poses, and standard error of the mean code. (a),(b),(c) are generated using DADL with the base dictionary $\mathbf{D_{10}}$. (d),(e),(f) are generated using Tensorfaces.

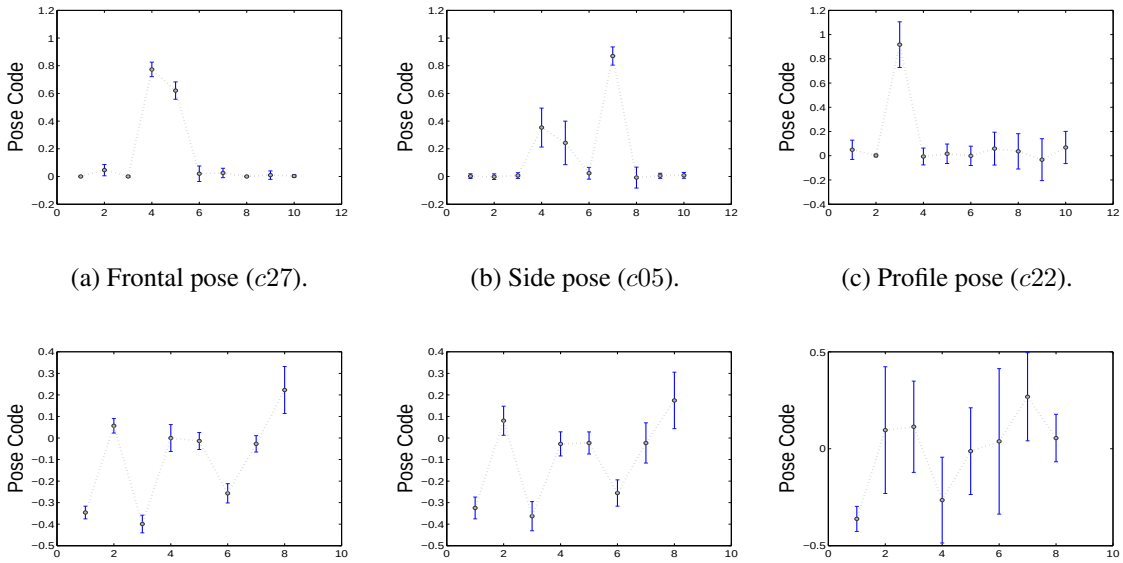


(a) Frontal pose ($c27$). (b) Side pose ($c05$). (c) Profile pose ($c22$).



(d) Frontal pose ($c27$). (e) Side pose ($c05$). (f) Profile pose ($c22$).

Figure 5.12: Mean illumination code of illumination condition $f1$ over 68 subjects in each of the three testing poses, and standard error of the mean code. (a),(b),(c) are generated using DADL with the base dictionary $\mathbf{D_{10}}$. (d),(e),(f) are generated using Tensorfaces.



(a) Frontal pose ($c27$). (b) Side pose ($c05$). (c) Profile pose ($c22$).

(d) Frontal pose ($c27$). (e) Side pose ($c05$). (f) Profile pose ($c22$).

Figure 5.13: Mean pose code of subject $s1$ over 21 illumination conditions for each of the three testing poses, and standard error of the mean code. (a),(b),(c) are generated using DADL with the base dictionary $\mathbf{D_{10}}$. (d),(e),(f) are generated using Tensorfaces.

Chapter 6

Directions for Future Work

In this chapter, we outline several potential directions in which the problems addressed in this dissertation can be explored further.

6.1 Unsupervised Domain Adaptive Dictionary Learning

Domain Adaptation discussed in this dissertation assumes correspondence between the source and target data. Unsupervised domain adaptation is a more challenging problem where no correspondence information is assumed across domains. Unsupervised DA is a more realistic setting frequently seen in real-life applications. For instance, face recognition models trained on faces collected under constrained conditions may not easily generalize to test data collected in unconstrained environments, where it is difficult to get labeled data for all possible variations.

We present here some initial considerations on unsupervised domain adaptation using dictionary-based methods for object recognition. This preliminary approach based on generating a set of intermediate domains which smoothly connect the source and target domains such that they correspond to the solutions of an optimization problem. This approach allows the synthesis of data associated with the intermediate domains. The intermediate domain data is used to build a classifier for recognition under domain shift. This initial approach will be further explored in future, for example additional geometry

constraints while generating intermediate domain observations.

6.1.1 Initial Considerations on Unsupervised DADL

Sparse representations are representations known for their succinct representation of stimuli and are able to represent high level patterns in the input signals. Let $\mathbf{X_s} \in \mathbb{R}^{n \times N_s}$ and $\mathbf{X_t} \in \mathbb{R}^{n \times N_t}$ be the data instances from the source and target domain respectively, where n is the dimension of the data instance and N_s , N_t denote the number of samples in the source and target domain. Given the source domain data $\mathbf{X_s}$, the standard dictionary learning technique aims to optimize the following cost function

$$\underset{\mathbf{D}_s, \mathbf{\Gamma}_s}{\arg\min} \|\mathbf{X}_s - \mathbf{D}_s \mathbf{\Gamma}_s\|_F^2, s.t. \forall i, \|\alpha_i\|_0 \leq T, \tag{6.1}$$

where $\mathbf{D_s}$ is the dictionary learned from $\mathbf{X_s}$, $\mathbf{\Gamma}_s = \{\alpha_i\}_{i=1}^{N_s}$ are the corresponding sparse codes for $\mathbf{X_s}$, and T is the sparsity level. We use the K-SVD algorithm to train the reconstructive dictionary $\mathbf{D_s}$. The resulting sparse codes provide a good feature representation, and we train a linear SVM classifier $\mathcal{C}$ from $\mathbf{\Gamma}_s$.

One can expect that the atoms in $\mathbf{D_s}$ are not necessarily optimal for the target domain data $\mathbf{X_t}$. Directly decomposing the target domain data $\mathbf{X_t}$ with $\mathbf{D_s}$ will result in a large reconstruction residue $\mathbf{J_s}$, and the corresponding classifier $\mathcal{C}$ will most likely perform unsatisfactorily on $\mathbf{X_t}$.

A feature representation which is preserved across different domains is an important factor for successful domain adaptation. We design sparse codes as an invariant feature across the source, the target, and $K-1$ intermediate domains, which leads to the following

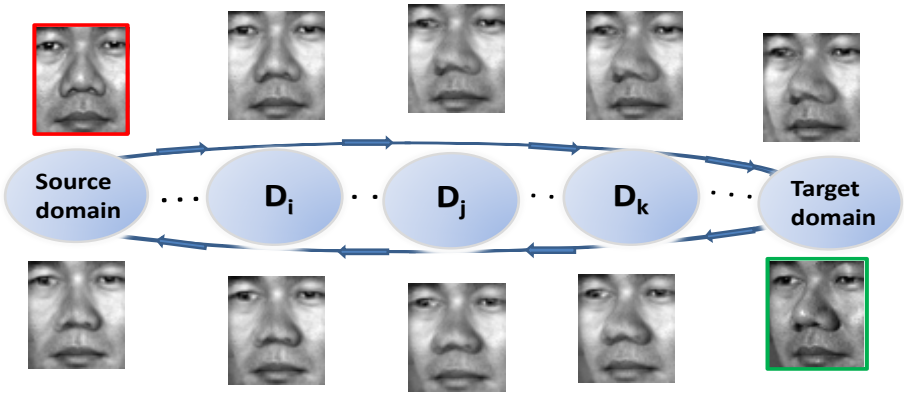


Figure 6.1: Given labeled data in the source domain and unlabeled data in the target domain, we propose an iterative dictionary learning procedure to learn a set of intermediate domains. We then generate corresponding intermediate observations associated with the intermediate domains.

objective function

$$\arg\min_{\mathbf{D}_k,\mathbf{\Gamma}} \sum_{k=0}^{K} \|\mathbf{X}_k - \mathbf{D}_k\mathbf{\Gamma}\|_F^2, s.t. \forall i, \|\alpha_i\|_0 \leq T, \tag{6.2}$$

For simplicity, we denote here the source domain and the target domain as the 0^{th} and the K^{th} domains respectively. In (6.2), $\mathbf{X}_k = \{x_{k,i}\}_{i=1}^{N_s}$ are the intermediate data in the k^{th} intermediate domain generated from the source data, $\mathbf{\Gamma} = \{\alpha_i\}_{i=1}^{N_s}$ are the domain invariant sparse codes. $\mathbf{D_0}$, $\{\mathbf{D_k}\}_{\mathbf{k=1}}^{\mathbf{K-1}}$, and $\mathbf{D_K}$ represent the dictionary associated with the source, intermediate and the target domains.

If intermediate data $\{\mathbf{X_k}\}_{\mathbf{k=1}}^{\mathbf{K-1}}$ are observed, (6.2) becomes similar to the problem discussed in [98]. However, as we do not assume the availability of $\{\mathbf{X_k}\}_{\mathbf{k=1}}^{\mathbf{K-1}}$, the objective function (6.2) is highly under-constrained. Hence, we propose the approach outlined in Algorithm 8 to approximately estimate $\{\mathbf{D_k}\}_{\mathbf{k=1}}^{\mathbf{K}}$. The intuition behind this algorithm is that: during each iteration, the reconstruction residue $\mathbf{J_k}$ gives an estimate of the gap between the current k^{th} intermediate domain and the target domain. The residual $\mathbf{J_k}$ pro-

vides gradient descent estimate for the next intermediate dictionary . In Algorithm 8, the learning rate λ is chosen to satisfy the smoothness assumption of the transition path, i.e., dictionaries $\mathbf{D_k}$ and $\mathbf{D_{k+1}}$ for adjacent domains do not change abruptly. This procedure is repeated until the reconstruction residue $\mathbf{J_k}$ is below the threshold δ. The final dictionary $\mathbf{D_K}$ will approximate the target data $\mathbf{X_t}$. We empirically observe the convergence of our algorithm in all our experiments.

Input: Dictionary $\mathbf{D_s}$ learned from the source data, target data $\mathbf{X_t}$, sparsity level T, threshold δ, learning rate λ

Output: Dictionaries $\{\mathbf{D_k}\}_{\mathbf{k=0}}^{\mathbf{K-1}}$ for intermediate domains and $\mathbf{D_K}$ for the target domain.

begin

1. Initialize $\mathbf{D_0}$:

   $\mathbf{D_0} = \mathbf{D_s}, k = 0$

2. Sparse coding: decompose the target data with the estimate of the dictionary $\mathbf{D_k}$ for the current intermediate domain:

   $\arg\min_{\mathbf{\Gamma}_k} \|\mathbf{X}_t - \mathbf{D}_k\mathbf{\Gamma}_k\|_F^2, s.t. \forall i, \|\alpha_i\|_0 \leq T$;

3. Compute the reconstruction residue:

   $\mathbf{J_k} = \mathbf{X_t} - \mathbf{D_k}\mathbf{\Gamma_k}$;

4. If $||\mathbf{J_k}||_\mathbf{F}^\mathbf{2} < \delta$, set $\mathbf{D_t} = \mathbf{D_k}$, return $\{\mathbf{D_k}\}_{\mathbf{k=0}}^{\mathbf{K}}$;

5. Get an estimate of the dictionary $\mathbf{D_{k+1}}$ for the next domain

   $\mathbf{D_{k+1}} = \mathbf{D_k} + \lambda \mathbf{J_k}\mathbf{\Gamma_k^T}(\mathbf{\Gamma_k}\mathbf{\Gamma_k^T})^{-1}$;

6. $k = k + 1$, go to step 2 ;

end

Algorithm 8: Algorithm to generate dictionaries for intermediate domains.

Next, given the set of dictionaries $\{\mathbf{D_k}\}_{\mathbf{k=0}}^{\mathbf{K}}$, we are able to generate the intermediate data from the source data $\mathbf{X_s}$, i.e., the approximated observations in the intermediate

domains. We first decompose the source data $\mathbf{X_s}$ with $\mathbf{D_0}$ to obtain $\mathbf{\Gamma_0}$, and obtain the intermediate data as $\{\mathbf{X_k}\}_{\mathbf{k=1}}^{\mathbf{K}} = \{\mathbf{D_k\Gamma_0}\}_{\mathbf{k=1}}^{\mathbf{K}}$. During each iteration, the intermediate data $\mathbf{X_k}$ are updated pertaining to the direction of the transition path, which is represented by $\mathbf{D_{k+1}} - \mathbf{D_k}$. The resulting sparse codes $\{\mathbf{\Gamma_k}\}_{\mathbf{k=0}}^{\mathbf{K}}$ appear to be consistent across the intermediate domains. Similarly, by traveling along the transition path in the reverse direction, we can generate intermediate data from the target data in a similar way as $\{\mathbf{X_k}\}_{\mathbf{k=1}}^{\mathbf{K}} = \{\mathbf{D_k\Gamma_K}\}_{\mathbf{k=1}}^{\mathbf{K}}$.

We will investigate two approaches for classifying target data based on the transition path represented in one of two schemes. First, a DA Classifier Invariant Codes (DAC-IC) approach: Given target data $\mathbf{X_t}$, and target domain dictionary $\mathbf{D_t}$, sparse codes are demonstrated to be invariant across different domains. Second, a Classifier Transition Path (DAC-TP) approach: We incorporate the rich information encoded in the transition path for improved classifier performance. As discussed, for each labeled source data $x_{s,i}$, we can generate a sequence of intermediate data $\{x_{s,i}^{(k)}\}_{k=1}^{K}$ as discussed above. Similarly, we obtain $\{x_{t,i}^{(k)}\}_{k=1}^{K}$ for each unlabeled target data $x_{t,i}$. We define the distance between the source data and the target data as the L2 norm between $x_{s,i}^{(k)}$ and $x_{t,i}^{(k)}$, and then a nearest neighbor classifier is used to infer the label of the unlabeled target data.

6.2 Structure-Preserved Sparse Decomposition for Actions

Extensive research has been conducted for modeling and recognition of human activities. Most existing work has focused on modeling and recognition of single person actions, including early approaches like 2D-templates model [108], hidden Markov model [109],

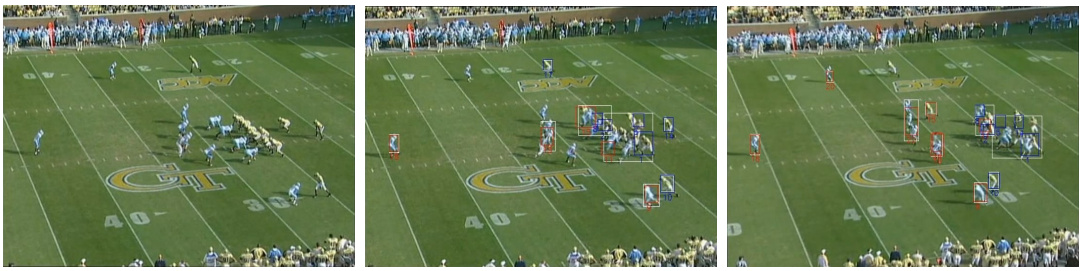
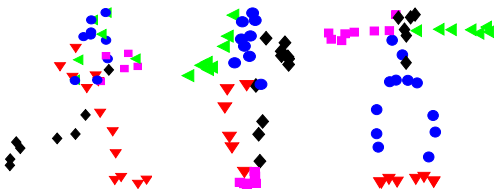


Figure 6.2: Sample frames of a football *Hitch* play video sequence

Group 1: {head, left shoulder,right shoulder}
Group 2: {left elbow, left wrist, left hand}
Group 3: {right elbow, right wrist, right hand}
Group 4: {hip, left knee, left ankle, left foot}
Group 5: {hip, right knee, right ankle, right foot}

(a) Human body motion groups

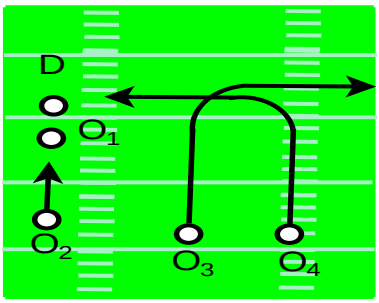


(b) Run (c) Salsa Dance (d) Indian Dance

Figure 6.3: Grouping for actions based on common motion. Trajectories at one time instant are shown. The resulting groups are of different shapes and colors.

or more recent approaches based on bag-of-words model [110], linear dynamical systems [111], etc.. Most of these approaches can not be directly applied to group activities due to the inherent difficulties in modeling inter-person interactions.

Group activities have been mostly modeled using Belief Networks [107], [112], [113], or other types of models like Petri nets [114]. Though many of these approaches are successful in modeling various group activity scenarios, they suffer from the following drawbacks. 1.) Manual specification of model structures is often required [114], [107]. Given the complex and unpredictable nature of human interactions, it is difficult to manually specify a comprehensive activity model. 2.) Models are often designed to handle specific types of activities, e.g., football plays [107], pairwise activities [113], etc. It



(a) Chalkboard sketch [107]

$\exists\ p_1, p_2, p_3, p_4, p_5,$
$\text{Offensive}(p_1) \wedge \text{Offensive}(p_2) \wedge \text{Offensive}(p_3)$
$\wedge\ \text{Offensive}(p_4) \wedge \text{Defensive}(p_5) \wedge \text{GoStraightUp}(p_3)$
$\wedge\ \text{TurnRight}(p_3) \wedge \text{GoStraightUp}(p_4) \wedge \text{TurnLeft}(p_4)$
$\wedge\ \text{GoStraightUp}(p_2) \wedge \text{StandStill}(p_1) \wedge \text{StandStill}(p_5)$
$\wedge\ \text{Sequential}\ (\text{GoStraightUp}(p_3),\ \text{TurnRight}(p_3))$
$\wedge\ \text{Sequential}\ (\text{GoStraightUp}(p_4),\ \text{TurnLeft}(p_4))$
$\wedge\ \text{Simultaneous}(\text{GoStraightUp}(p_3),\ \text{GoStraightUp}(p_4))$
$\wedge\ \text{Simultaneous}(\text{TurnRight}(p_3),\ \text{TurnLeft}(p_4))$
$\wedge\ \text{Simultaneous}(\text{GoStraightUp}(p_3),\ \text{GoStraightUp}(p_2))$
$\wedge\ \text{Simultaneous}(\text{GoStraightUp}(p_2),\ \text{StandStill}(p_1))$
$\wedge\ \text{Simultaneous}(\text{StandStill}(p_1),\ \text{StandStill}(p_5))$

(b) Semantic interpretation

Figure 6.4: The football *simple-p51curl* play

can be difficult for extensions to activities involving more persons or other scenarios. 3.) Techniques for matching entities in a video and entities in the model are often not carefully addressed [115], [107]. Given activities like football plays shown in Fig. 6.2, which involve 22 players, such entity correspondence problem can not be trivially handled.

To address the challenges above, we start with an approach similar to [116] which describes human actions or activities using entities and predicates. For individual actions, as shown in Fig. 6.3, body parts share common motions due to human articulation constraints. For example, the *running* action is interpreted as $\exists\ b_1, b_2,$ *LeftLeg*(b_1) $\wedge$ *RightLeg*(b_2) $\wedge$ *RunMotion*(b_1) $\wedge$ *RunMotion*(b_2) $\wedge$ *Simultaneous(RunMotion*(b_1), *RunMotion*(b_2)). For structured group activities, collaborative players can have correlated motions. A *Simple-p51curl* football play activity involving 4 offensive players and 1 defensive player in Fig. 6.4a, which can be sketched by a coach, is represented in Fig. 6.4b, where we list the temporal constraints among motions. One can also incorporate spatial constraints such as orientation and distance.

From the above examples, one can notice that a formula for each action or activity

consists of three types of predicates describing entities, e.g., *Offensive(p_3)*, atomic motions, e.g., *TurnRight(p_3)*, and pairwise motion constraints, e.g., *Simultaneous(TurnRight(p_3), TurnLeft(p_4))*. It is noted that an entity here is defined as any moving person, body part, or object. A group of people moving in a coordinated way sometimes can be as a whole considered as an entity. Though only we limit ourselves to pairwise interactions, higher order relationships can be introduced using more complex models.

The above formulation faces the following two specific problems for activity modeling. First, an MRF constructed from an MLN that models activities can easily contains a large number of nodes. During grounding, an existential quantifier in MLN is expended over the entire entity domain to obtain a disjunction of the original formula. For example, an activity is given as a formula $\exists p_1, p_2$, *TurnRight(p_1)*$\wedge$ *TurnLeft(p_2)*, and two entities P_1 and P_2 are detected in the video. Since it is typically difficult to associate entities in a video with entities specified in the formula, the grounded formula will be *(TurnRight(P_1)*$\wedge$ *TurnLeft(P_2))* $\vee$ *(TurnRight(P_2)*$\wedge$ *TurnLeft(P_1))*. Thus, using the MLN exhaustive grounding scheme, we can ground extremely complex MRF structure from an MLN, e.g., a network corresponds to the disjunction of 22! conjunctive clauses for a 22 player football activity. Second, Each predicate can require separate manual modeling to encode the knowledge [116], which is a tedious task. Therefore, we seek for a scheme to perform structure-preserved decomposition for complex actions. With such structure-preserved decomposition, a complex activity can be described as Fig. 6.4b, i.e., a set of semantic units connected using spatial and temporal constraints.

6.3 Alignment Invariant Sparse Representation

Sparse representation-based approaches are known to be sensitive to misalignment. For example, the sparse representation-based classification (SRC) method [79] has demonstrated the state of the art recognition performance despite severe occlusion or corruption. The main idea of SRC is that the nonzero coefficients should concentrate on the training samples with the same class label as the test sample. However, as shown in Fig. 6.5, SRC does not deal well with misalignments between the test and training images. Even small registration error against the training images, the sparse representation obtained for testing images can become non-informative. We propose to study the problem of alignment invariant sparse representation.

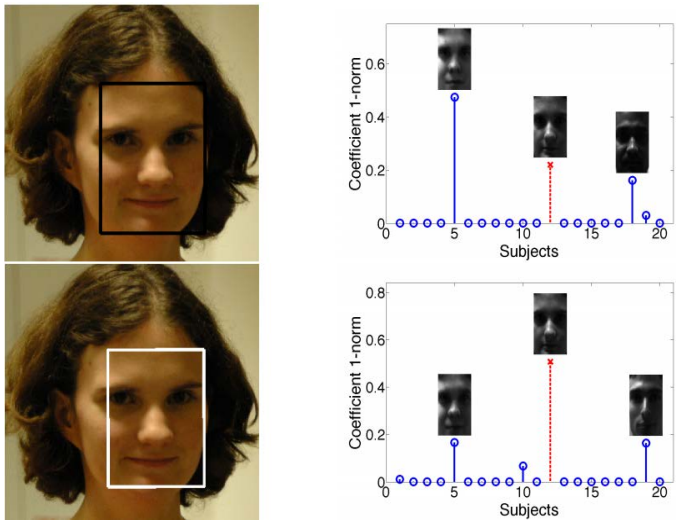


Figure 6.5: Effects of misalignments on recognition using sparse representation [9]. Top: The input face is from Viola and Jones' face detector. Bottom: The input face is well aligned to the training data.

Bibliography

[1] J. Liu and M. Shah. Learning human actions via information maximization. In *IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Anchorage, Alaska*, June 2008.

[2] I. Laptev, M. Marszałek, C. Schmid, and B. Ronfeld. Learning realistic human actions from movies. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Anchorage, Alaska*, June 2008.

[3] J. A. Tropp, A. C. Gilbert, and M. J. Strauss. Algorithms for simultaneous sparse approximation. part i: Greedy pursuit. *Signal Processing*, 86:572–588, 2006.

[4] Qiang Qiu, Zhuolin Jiang, and Rama Chellappa. Sparse dictionary-based representation and recognition of action attributes. In *Proc. Intl. Conf. on Computer Vision, Barcelona, Spain*, Nov. 2011.

[5] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical analysis on manifolds and its applications to video analysis. *Video Search and Mining, Studies in Computational Intelligenceg*, 287:115–144, 2010.

[6] M. Alex O. Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proc. European Conf. on Computer Vision, Copenhagen, Denmark*, May 2002.

[7] M. Alex O. Vasilescu and Demetri Terzopoulos. Multilinear image analysis for facial recognition. In *Proc. Intl. Conf. on Patt. Recn., Quebec, Canada*, Aug. 2002.

[8] Carlos D. Castillo and David W. Jacobs. Using stereo matching for 2-d face recognition across pose. *IEEE Trans. on Patt. Analysis and Mach. Intell.*, 31:2298–2304, 2009.

[9] Andrew Wagner, John Wright, Arvind Ganesh, Zihan Zhou, Hossein Mobahi, and Yi Ma. Toward a practical face recognition system: Robust alignment and illumination by sparse representation. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 34(2):372–386, 2012.

[10] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generatlization. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., San Francisco. CA*, June 2010.

[11] C. Lampert, H. Nickisch, and S. Harmerling. Learning to detect unseen object classes by between-class attribute transfer. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Miami, FL*, June 2009.

[12] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Rec., Miami, FL*, June 2009.

[13] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., San Francisco. CA*, June 2010.

[14] X. Yu and Y. Aloimonos. Attribute-based transfer learning for object categorization with zero/one training example. In *Proc. European Conf. on Computer Vision, Crete, Greece*, Sep. 2010.

[15] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. the MIT Press, 2006.

[16] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J.W. Vaughan. A theory of learning fromdifferent domains. *Machine Learning*, 79:151–175, May 2010.

[17] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. Knowledge and Data Engineering*, 22:1345–1359, October 2010.

[18] Ralph Gross, Simon Baker, Iain Matthews, and Takeo Kanade. Face recognition across pose and illumination. In *Handbook of Face Recognition*. Springer-Verlag, 2004.

[19] C.D. Castillo and D.W. Jacobs. Wide-baseline stereo for face recognition with large pose variation. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Colorado Springs, CO*, June 2011.

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

[21] K. Engan, S. Aase, and J. Husϕy. Frame based signal compression using method of optimal directions (MOD). In *IEEE Intern. Symp. Circ. Syst., Orlando, FL*, May 1999.

[22] J. Wright, M. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 31(2):210–227, 2009.

[23] J. Mairal, F. Bach, J. Pnce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Anchorage, Alaska*, 2008.

[24] Q. Zhang and B. Li. Discriminative k-SVD for dictionary learning in face recognition. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., San Francisco, CA*, June 2010.

[25] D. Pham and S. Venkatesh. Joint learning and dictionary construction for pattern recognition. In *IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Anchorage, Alaska*, June 2008.

[26] Zhuolin Jiang, Zhe Lin, and Larry S. Davis. Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Colorado springs, CO*, June 2011.

[27] K Etemad and Rama Chellappa. Separability-based multiscale basis selection and feature extraction for signal and image classification. *IEEE Trans. on Image Process.*, 7(10):1453–1465, 1998.

[28] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *International Conference on Machine Learning, Alberta, Canada*, 2004.

[29] L. Wang, L. Zhou, and C. Shen. A fast algorithm for creating a compact and discriminative visual codebook. In *Proc. European Conf. on Computer Vision, Marseiiles, France*, Oct. 2008.

[30] S. Lazebnik and M. Raginsky. Supervised learning of quantizer codebooks by information loss minimization. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 31(7):1294–1309, 2009.

[31] N. Slonim and N. Tishy. Document clustering using word clusters via the information bottleneck method. In *International ACM SIGIR Conference, Athens, Greece*, July 2000.

[32] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *Neural Information Processing Systems, Vancouver, Canada*, Dec. 2008.

[33] C. Thurau and V. Hlavac. Pose primitive based human action recognition in videos or still images. In *IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Anchorage, Alaska*, June 2008.

[34] D. Weinland and E. Boyer. Action recognition using exampler-based embedding. In *IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Anchorage, Alaska*, June 2008.

[35] A. Elgammal, V. Shet, Y. Yacoob, and L. Davis. Learning dynamics for exemplar-based geature recognition. In *IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Madison, WI*, June 2003.

[36] Y. Li, C. Fermuller, and Y. Aloimonos. Learning shift-invariant sparse representation of actions. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., San Francisco, CA*, June 2010.

[37] F. Rodriguez and G. Sapiro. Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaries. *Tech. Report, University of Minnesota*, Dec. 2007.

[38] K. Huang and S. Aviyente. Sparse representation for signal classification. In *Neural Information Processing Systems, Vancouver, Canada*, Dec. 2007.

[39] Z. Lin, Z. Jiang, and L. Davis. Recognizing actions by shape-motion prototype trees. In *Proc. Intl. Conf. on Computer Vision, Kyoto, Japan*, Oct. 2009.

[40] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., San Diego, CA*, June 2005.

[41] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proc. Intl. Conf. on Computer Vision, Nice, France*, 2003.

[42] Sreemanananth Sadanand and Jason J. Corso. Action bank: A high-level representation of activity in video. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Providence, RI*, June 2012.

[43] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

[44] N. Shroff, P. Turaga, and R. Chellappa. Video precis: Hightlighting diverse aspects of videos. *IEEE Transactions on Multimedia*, 12(8):853–868, 2010.

[45] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *Proc. Intl. Conf. on Computer Vision, Beijing, China*, Oct. 2005.

[46] Mikel D. Rodriguez, Javed Ahmed, and Mubarak Shah. Action mach: A spatio-temporal maximum average correlation height filter for action recognition. In *IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Anchorage, Alaska*, 2008.

[47] UCF50 dataset. `http://www.cs.ucf.edu/vision/public\_html/data.html`.

[48] L. Yeffet and L. Wolf. Local trinary patterns for human action recognition. In *Proc. Intl. Conf. on Computer Vision, Kyoto, Japan*, Nov. 2009.

[49] M. Varma and B. R. Babu. More generality in efficient multiple kernel learning. In *International Conference on Machine Learning, Montreal, Canada*, June 2009.

[50] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference, London*, Sep. 2009.

[51] Q. Le, W. Zou, S. Yeung, and A. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Proc. IEEE Computer Society Cnf. on Computer Vision and Patt. Recn., Colorado springs, CO*, June 2011.

[52] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., San Francisco. CA*, June 2010.

[53] X. Wu, D. Xu, L. Duan, and J. Luo. Action recognition using context and appearance distribution features. In *Proc. IEEE Computer Society Cnf. on Computer Vision and Patt. Recn., Colorado springs, CO*, June 2011.

[54] L. Latecki, R. Lakamper, and T. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Hilton Head, SC*, June 2000.

[55] R. Rubinstein, A.M. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, june 2010.

[56] J. Wright, Yi Ma, J. Mairal, G. Sapiro, T.S. Huang, and Shuicheng Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6):1031–1044, june 2010.

[57] M. Elad, M.A.T. Figueiredo, and Yi Ma. On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE*, 98(6):972 –982, June 2010.

[58] Vishal M. Patel and Rama Chellappa. Sparse representations, compressive sensing and dictionaries for pattern recognition. In *First Asian Conference on Pattern Recognition (ACPR), Beijing, China*, Dec. 2011.

[59] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comp.*, 20(1):33–61, 1998.

[60] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. *Proc. 27th Asilomar Conference on Signals, Systems and Computers*, pages 40–44, Nov. 1993.

[61] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. Info. Theory*, 50(10):2231–2242, Oct. 2004.

[62] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[63] V. M. Patel, T. Wu, S. Biswas, P. J. Phillips, and R. Chellappa. Dictionary-based face recognition under variable lighting and pose. *IEEE Transactions on Information Forensics and Security*, 7(3):954–965, June 2012.

[64] K. Etemand and R. Chellappa. Separability-based multiscale basis selection and feature extraction for signal and image classification. *IEEE Trans. on Image Processing*, 7(10):1453–1465, Oct. 1998.

[65] M. Yang, X. Feng L. Zhang, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. In *Proc. Intl. Conf. on Computer Vision, Barcelona, Spain*, Nov. 2011.

[66] E. Kokiopoulou and P. Frossard. Semantic coding by supervised dimensionality reduction. *IEEE Trans. Multimedia*, 10(5):806–818, Aug. 2008.

[67] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 34(4):791 –804, April 2012.

[68] J. Mairal, F. Bach, J. Pnce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Anchorage*, 2008.

[69] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *Neural Information Processing Systems, Vancouver, Canada*, Dec. 2008.

[70] M. E. Hellman and J. Raviv. Probability of error, equivocation, and the Chernoff bound. *IEEE Trans. on Info. Theory*, 16:368–372, 1979.

[71] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[72] Kari Torkkola. Feature extraction by non parametric mutual information maximization. *JMLR*, 3:1415–1438, Mar. 2003.

[73] J. Kapur. *Measures of information and their applications*. Wiley, 1994.

[74] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Ilumination cone models for face recognition under variable lighting and pose. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 23(6):643–660, June 2001.

[75] USPS handwritten digit database. `http://www-i6.informatik.rwth-aachen.de/~keysers/usps.html`.

[76] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Computer Society Conf. on Computer Vision and Patt. Recn., New York, NY*, June 2006.

[77] Jianchao Yang, Kai Yu, Yihong Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Rec., Miami, FL*, June 2009.

[78] Jan C. Gemert, Jan-Mark Geusebroek, Cor J. Veenman, and Arnold W. Smeulders. Kernel codebooks for scene categorization. In *Proc. European Conf. on Computer Vision, Marseiiles, France*, Oct. 2008.

[79] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 31(2):210–227, 2009.

[80] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., San Francisco*, June 2010.

[81] S. Gong, S. J. McKenna, and A. Psarrou. *Dynamic vision from images to face recognition*. Imperial College Press, 2000.

[82] Thomas Vetter and Tomaso Poggio. Linear object classes and image synthesis from a single example image. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 19(7):733–742, 1997.

[83] D. Beymer, A. Shashua, and T. Poggio. Example-based image analysis and synthesis. *Artificial Intelligence Laboratory A.I. Memo No. 1431*, 19(121), 1993.

[84] David Beymer and Tomaso Poggio. Face recognition from one example view. *Artificial Intelligence Laboratory A.I. Memo No. 1536*, 19(121), 1995.

[85] P. Lancaster and K. Salkauskas. *Curve and Surface Fitting*. Academic Press, 1990.

[86] Alan Edelman, Tomás A. Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Analysis and Applications*, 20:303–353, April 1999.

[87] Ehsan Elhamifar and Ren Vidal. Sparse subspace clustering. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Miami, FL,*, June 2009.

[88] Luis Machado and F. Silva Leite. Fitting smooth paths on riemannian manifolds. *Int. J. Appl. Math. Stat.*, 4:25–53, 2006.

[89] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 25(12):1615 –1618, Dec. 2003.

[90] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 23(6):643–660, June 2001.

[91] M.A. Turk and A.P. Pentland. Face recognition using eigenfaces. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Maui, Hawaii*, June 1991.

[92] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, 2010.

[93] Hal Daumé, III and Daniel Marcu. Domain adaptation for statistical classifiers. *J. Artif. Int. Res.*, 26(1):101–126, May 2006.

[94] Ali Farhadi and Mostafa Kamali Tabrizi. Learning to recognize activities from the wrong view point. In *Proc. European Conference on Computer Vision, Marseille, France*, Oct. 2008.

[95] R. Gopalan, R. Li, , and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Proc. Intl. Conf. on Computer Vision, Barcelona, Spain*, Nov. 2011.

[96] Sung Won Park and Marios Savvides. Individual kernel tensor-subspaces for robust face recognition: A computationally efficient tensor framework without requiring mode factorization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(5):1156–1166, 2007.

[97] Joshua B. Tenenbaum and William T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000.

[98] Qiang Qiu, Vishal Patel, Pavan Turaga, and Rama Chellappa. Domain adaptive dictionary learning. In *Proc. European Conference on Computer Vision, Florence, Italy*, Oct. 2012.

[99] J.A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Trans. on Information Theory*, 50:2231 – 2242, 2004.

[100] R. Chellappa, C.L. Wilson, and S. Sirohey. Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, 83:705–740, May 1995.

[101] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. on Patt. anal. and Mach. Intell.*, 19(7):711–720, July 1997.

[102] Ronen Basri and David W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Trans. on Patt. anal. and Mach. Intell.*, 25(2):218–233, February 2003.

[103] Zhanfeng Yue, Wenyi Zhao, and Rama Chellappa. Pose-encoded spherical harmonics for face recognition and synthesis using a single image. *EURASIP Journal on Advances in Signal Processing*, 2008(65), January 2008.

[104] R. Ramamoorthi and P. Hanrahan. A signal-processing framework for reflection. *ACM Transactions on Graphics*, 23(4):1004–1042, Oct 2004.

[105] Y. Tanabe, T. Inui, and Y. Onodera. *Group Theory and Its Applications in Physics*. Springer, Berlin, Germany, 1990.

[106] Haichao Zhang, Jianchao Yang, Yanning Zhang, and Thomas Huang. Close the loop: Joint blind image restoration and recognition with sparse representation prior. In *Proc. Intl. Conf. on Computer Vision, Barcelona, Spain*, Nov. 2011.

[107] Stephen S. Intille and Aaron F. Bobick. Recognizing planned multiperson action. *Comput. Vis. Image Underst.*, 81(3):414–445, March 2001.

[108] Ramprasad Polana and Randal C. Nelson. Detection and recognition of periodic, nonrigid motion. *Int. J. Comput. Vision*, 23(3):261–282, 1997.

[109] J Yamato, J Ohya, and K Ishii. Recognizing human action in time-sequential images using hidden markov model. *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Champaign, IL*, June 1992.

[110] Juan Carlos Niebles, Hongcheng Wang, and Li Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *Int. J. Comput. Vision*, 79(3):299–318, 2008.

[111] P. K. Turaga, A.Veeraraghavan, and R. Chellappa. From videos to verbs: Mining videos for activities using a cascade of dynamical systems. In *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Minneapolis, Minnesota*, June 2007.

[112] Asaad Hakeem and Mubarak Shah. Learning, detection and representation of multi-agent events in videos. *Artificial Intelligence*, pages 586–605, 2007.

[113] Yue Zhou, Shuicheng Yan, and Thomas S. Huang. Pair-activity classification by bi-trajectories analysis. *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn.,Anchorage, Alaska*, June 2008.

[114] M. Albanese, V. Moscato, R. Chellappa, A. Picariello, V. S. Subrahmanian, P. Turaga, and O. Udrea. A constrained probabilistic petri-net framework for human activity detection in video. *IEEE Transactions on Multimedia*, 10(8):1429 –1443, dec. 2008.

[115] Ruonan Li, R. Chellappa, and S.K. Zhou. Learning multi-modal densities on discriminative temporal interaction manifold for group activity recognition. *Proc. IEEE Computer Society Conf. on Computer Vision and Patt. Recn., Miami, FL*, June 2009.

[116] Son D. Tran and Larry S. Davis. Event modeling and recognition using markov logic networks. In *Proc. European Conf. on Computer Vision, Marseiiles, France*, 2008.