Kernelized Supervised Dictionary Learning

by

Mehrdad Jabbarzadeh Gangeh

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2013

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The representation of a signal using a *learned* dictionary instead of *predefined* operators, such as wavelets, has led to state-of-the-art results in various applications such as denoising, texture analysis, and face recognition. The area of dictionary learning is closely associated with sparse representation, which means that the signal is represented using *few atoms* in the dictionary. Despite recent advances in the computation of a dictionary using fast algorithms such as K-SVD, online learning, and cyclic coordinate descent, which make the computation of a dictionary from millions of data samples computationally feasible, the dictionary is mainly computed using unsupervised approaches such as k-means. These approaches learn the dictionary by minimizing the reconstruction error without taking into account the category information, which is not optimal in classification tasks.

In this thesis, we propose a supervised dictionary learning (SDL) approach by incorporating information on class labels into the learning of the dictionary. To this end, we propose to learn the dictionary in a space where the dependency between the signals and their corresponding labels is maximized. To maximize this dependency, the recently-introduced Hilbert Schmidt independence criterion (HSIC) is used. The learned dictionary is compact and has closed form; the proposed approach is fast. We show that it outperforms other unsupervised and supervised dictionary learning approaches in the literature on real-world data.

Moreover, the proposed SDL approach has as its main advantage that it can be easily kernelized, particularly by incorporating a data-driven kernel such as a compression-based kernel, into the formulation. In this thesis, we propose a novel compression-based (dis)similarity measure. The proposed measure utilizes a 2D MPEG-1 encoder, which takes into consideration the spatial locality and connectivity of pixels in the images. The proposed formulation has been carefully designed based on MPEG encoder functionality. To this end, by design, it solely uses P-frame coding to find the (dis)similarity among patches/images. We show that the proposed measure works properly on both small and large patch sizes on textures. Experimental results show that by incorporating the proposed measure as a kernel into our SDL, it significantly improves the performance of a supervised pixel-based texture classification on Brodatz and outdoor images compared to

other compression-based dissimilarity measures, as well as state-of-the-art SDL methods. It also improves the computation speed by about 40% compared to its closest rival.

Eventually, we have extended the proposed SDL to multiview learning, where more than one representation is available on a dataset. We propose two different multiview approaches: one fusing the feature sets in the original space and then learning the dictionary and sparse coefficients on the fused set; and the other by learning one dictionary and the corresponding coefficients in each view separately, and then fusing the representations in the space of the dictionaries learned. We will show that the proposed multiview approaches benefit from the complementary information in multiple views, and investigate the relative performance of these approaches in the application of emotion recognition.

Acknowledgements

All the praises and thanks are to God who gave me the strength and knowledge to accomplish this research.

Writing this thesis was not possible without support of my highly esteemed supervisors Dr. Mohamed Kamel and Dr. Ali Ghodsi. They both assisted me through many interesting technical conversations, offered guidance, and made suggestions that helped make my research work better. I would also like to express my gratitude to my Ph.D. committee: Dr. Ling Guan, Dr. Paul Fieguth, Dr. Otman Basir, and Dr. Zhou Wang, for providing valuable feedback and comments on my thesis.

I am very grateful for support and suggestions from many colleagues at University of Waterloo, especially in the Center for Pattern Analysis and Machine Intelligence (CPAMI), including Aya Sayedelahl, Pouria Fewzee, Ahmed Farahat, Pooyan Khajehpour, Rodrigo Araujo, Mike Miao Yun-Qian, Abbas Ahmadi, Michael Diu, Amir Hossein Shabani, Hossein Parsaei, Babak Alavi-Kia, Sepideh Seifzadeh, Kaushik Roy, Farook Sattar, Fatemeh Dorri, Nabil Drawil, and Yibo Zhang.

This research was financially supported by the Natural Sciences and Engineering Research Council (NSERC) and also the Ontario Ministry of Training, Colleges and Universities. This support made it also possible for me to visit the Pattern Recognition Lab at Delft University of Technology in the Netherlands, where I was thrilled by their high-quality research work. I especially enjoyed having technical discussions with Dr. Robert P.W. Duin, Marco Loog, David Tax, and Laurens van der Maaten.

I gratefully acknowledge the joint work with: Lauge Sørensen and Marleen de Bruijne from the Dept. of Computer Science, Copenhagen University, Denmark; Saher B. Shaker from the Department of Respiratory Medicine, Gentofte University Hospital, Hellerup, Denmark; and Ali Sadegji Naini and Dr. Gregory Czarnota from the Dept. of Radiation Oncology and Imaging Research, Sunnybrook Health Sciences Center, Canada.

A research work cannot be accomplished without highly-qualified support staff. I'd like to express my appreciation to Ms. Rosalind Klein, the secretary of the CPAMI, and all other supporting staff at the ECE for their caring and kind administrative support.

I thank very much my parents for their unconditional love and kindness, and my brother and sisters for their encouragement and good wishes during my Ph.D.

Last but not least, special thanks to my wife, Maryam, for her love, patience, encouragement, and support during my research, and to my sons, Ali and Iman, for their understanding and also for giving joy and happiness to my life.

Dedication

This thesis is dedicated to my wife, Maryam.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

1D	One Dimensional
2D	Two Dimensional
AIB	Agglomerative Information Bottleneck
AVEC	Audio/Visual Emotion Challenge
BL	Bilinear
BP	Basis Pursuit
BSS	Blind Source Separation
CCA	Canonical Correlation Analysis
CFA	Cross-Modal Factor Analysis
CK+	Extended Cohn-Kanade
DCT	Discrete Cosine Transform
DK-SVD	Discriminative K-SVD
DLSI	Dictionary Learning with Structured Incoherence
DLSR	Dictionary Learning and Sparse Representation
FCSC	Fully-Continuous Sub-Challenge
FDDL	Fisher Discrimination Dictionary Learning
FER	Facial Expression Recognition
FMF	Facial Measure Features
GMM	Gaussian Mixture Model
HNR	Harmonic-to-Noise Ratio
HSIC	Hilbert Schmidt Independence Criterion
IB	Information Bottleneck
ICA	Independent Component Analysis

JAFFE	Japanese Female Facial Expression
JPEG	Joint Photographic Experts Group
KL	Kullback-Leibler
LASSO	Least Absolute Shrinkage and Selection Operator
LBP	Local Binary Pattern
LLD	Low-Level Description
MAP	Maximum A Posteriori
MFCC	Mel Frequency Cepstrum Coefficients
MKL	Multiple Kernel Learning
MLE	Maximum Likelihood Estimation
MPEG	Moving Picture Experts Group
MV	Multiview
NCD	Normalized Compression Distance
NED	Normalized Entropy-rate Distance
NID	Normalized Information Distance
NNMF	Nonnegative Matrix Factorization
OMP	Orthogonal Matching Pursuit
RBF	Radial Basis Function
RCF	Randomized Clustering Forest
SAL	Sensitive Artificial Listener
SC	Sparse Coding
SCPO	Self-describing Content-based Pixel Ordering
SDL	Supervised Dictionary Learning
SDL-D	Supervised Dictionary Learning-Discriminative
SDL-G	Supervised Dictionary Learning-Generative
SDLM	Supervised Dictionary Learning Model
SED	Spectral Energy Distribution
SER	Speech Emotion Recognition
SIFT	Scale Invariant Feature Transform
SRC	Sparse Representation-based Classification
SV	Single View

SVM	Support Vector Machine
SVR	Support Vector Regression
UAV	Universal and Adapted Vocabulary
UVD	Universal Visual Dictionary
VAM	Vera Am Mittag

Chapter 1

Introduction

There are many mathematical models to describe data with varying degrees of success, among which dictionary learning and sparse representation (DLSR) has attracted the interest of many researchers in various fields. Dictionary learning and sparse representation are two closely-related topics that have roots in the decomposition of signals to some predefined bases, such as the Fourier transform. Representation of signals using predefined bases is based on the assumption that these bases are general enough to represent any kind of signal, however, recent research shows that learning the bases[1] from data, instead of using off-the-shelf ones, leads to state-of-the-art results in many applications [1]. In fact, what makes DLSR distinct from the representation using predefined bases is that first, the bases are learned here from the data, and second, only a few components in the dictionary are needed to represent the data (sparse representation). This latter attribute can also be seen in the decomposition of signals using some predefined bases such as wavelets [2].

[1] Here, the term basis is loosely used as the dictionary can be overcomplete, i.e., the number of dictionary elements can be larger than the dimensionality of the data, and its atoms are not necessarily orthogonal and can be linearly dependent.

1.1 Related Topics

The concept of dictionary learning and sparse representation originated in different communities attempting to solve different problems, which are given different names. Some of them are: sparse coding (SC), which was originated by neurologists as a model for simple cells in mammalian primary visual cortex [3]; independent component analysis (ICA), which was developed by researchers in signal processing to estimate the underlying hidden components of multivariate statistical data (refer to [4] for a review of ICA); least absolute shrinkage and selection operator (*lasso*), which was used by statisticians to find linear regression models when there are many more predictors than samples, where some constraints have to be considered to fit the model. In the *lasso*, one of the constraints introduced by Tibshirani was the ℓ_1-norm that led to sparse coefficients in the linear regression model [5]. Another technique which also leads to DLSR is nonnegative matrix factorization (NNMF), which aims to decompose a matrix to two nonnegative matrices, one of which can be considered to be the dictionary, and the other the coefficients [6]. In NNMF, usually both the dictionary and coefficients are sparse [6,7]. This list is not complete, and there are variants for each of the above techniques, such as blind source separation (BSS) [8], compressed sensing [9], basis pursuit (BP) [10], and orthogonal matching pursuit (OMP) [11,12]. It is beyond the scope of this thesis to include the description of all these techniques (interested readers can refer to [1,13–15] for reviews on dictionary learning and sparse representation).

The main results of all these research efforts is that a class of signals with sparse nature, such as images of natural scenes, can be represented using some *primitive elements* that form a dictionary, and that each signal in this class can be represented by using only a few elements in the dictionary, i.e., by a sparse representation. In fact, there are at least two ways in the literature to exploit sparsity [16]: first, using a linear/nonlinear combination of some predefined bases, e.g., wavelets [2]; second, using primitive elements in a learned dictionary, such as the techniques employed in SC or ICA. This latter approach is the focus of this thesis and has led to state-of-the-art results in various applications such as texture classification [17–19], face recognition [20–22], image denoising [23,24], biomedical tissue characterization [25–27], motion and data segmentation [28,29], data representation and column selection [30], and image super-resolution [31]. Figure 1.1 summarizes the topics

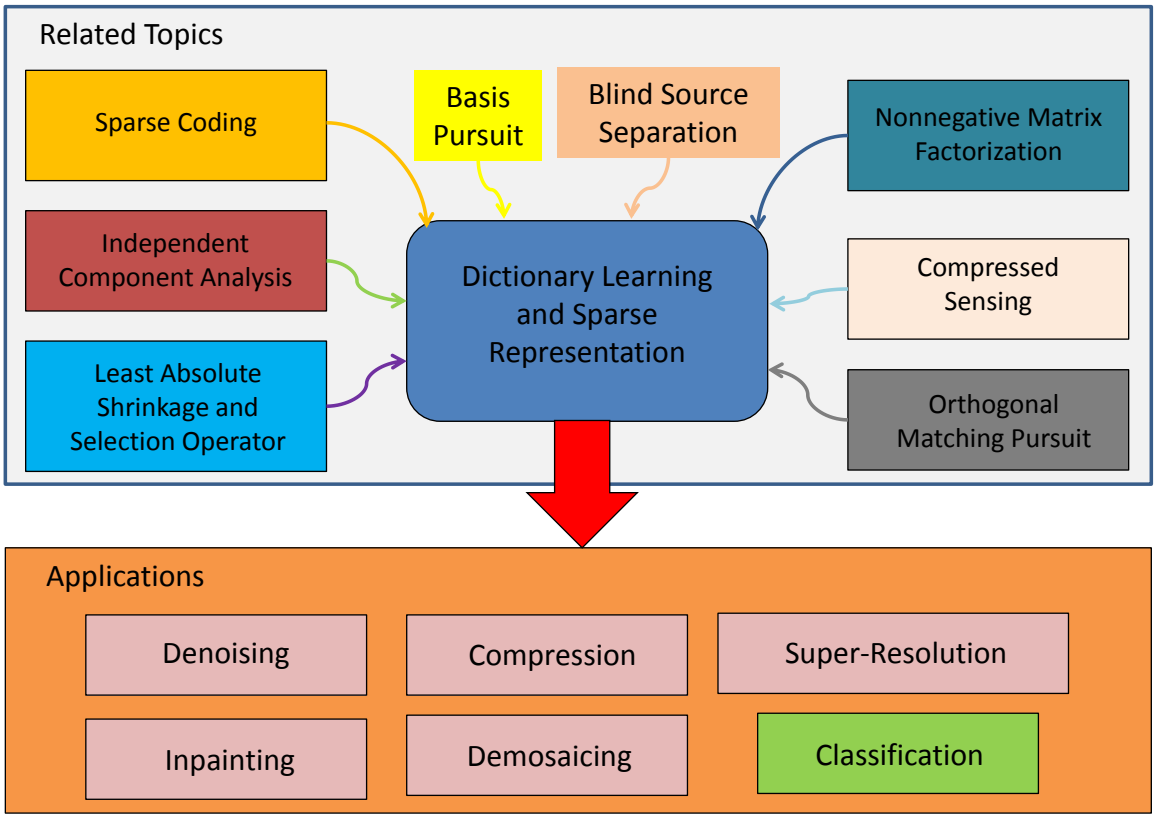


Figure 1.1: Topics related to and the applications of dictionary learning and sparse representation.

related to and the applications of dictionary learning and sparse representation.

1.2 Taxonomy of DLSR

One may categorize the various dictionary learning with sparse representation approaches proposed in the literature in different ways: one where the dictionary consists of predefined or learned bases as stated above, and the other based on the model used to learn the dictionary and coefficients. These models can be *generative* as used in the original formulation of SC [3], ICA [4], and NNMF [6]; *reconstructive* as in the *lasso* [5]; or discriminative such as

SDL-D (supervised dictionary learning-discriminative) in [16]. The two former approaches do not consider the class labels in building the dictionary, while the last one (i.e., the discriminative one) does. In other words, dictionary learning can be performed unsupervised or supervised, with the difference that in the latter, the class labels in the training set are used to build a more discriminative dictionary for the particular classification task in hand.

1.3 Objectives and Contributions

The main objectives of this thesis are as follows:

- To develop a supervised dictionary learning (SDL) algorithm by incorporating class labels into the learning of the dictionary;
- To design and incorporate a compression-based dissimilarity measure into the designed SDL as a kernel to further improve the discrimination power of the algorithm in subtle classification tasks;
- To extend the proposed SDL to multiview representations.

As the result of the research carried out, a novel supervised dictionary learning is proposed in this thesis by incorporating information on class labels into the learning of the dictionary. The dictionary is learned in a space where the dependency between the data and their corresponding labels is maximized. It is proposed to maximize this dependency by using the recently introduced Hilbert Schmidt independence criterion (HSIC) [32, 33]. Although supervised dictionary learning has been proposed by others, as will be reviewed in the next chapter, this work is different from the others in the following aspects:

1. The formulation is simple and straightforward;
2. The proposed approach introduces a closed form formulation for the computation of the dictionary. This is different from other approaches, in which the computation of dictionary and sparse coefficients has to be iteratively and often alternately performed, which causes high computational load;

3. The approach is very efficient in terms of dictionary size (compact dictionary). The results show that the proposed dictionary can produce significantly better results than other supervised dictionary methods for small dictionary sizes. An important special case is when the dictionary size is smaller than the dimensionality of data. This turns the learning of a dictionary whose size is usually larger than the dimensionality of the data, i.e., an *overcomplete* dictionary, into the learning of a *subspace*;

4. The proposed approach can be easily kernelized by incorporating a kernel into the formulation. For example, data-dependent kernels based on normalized compression distance (NCD) [34, 35], can be used in this kernelized SDL to further improve the discrimination power of the designed system. To the best of my knowledge, no other kernelized SDL approach has been proposed in the literature yet, and none of the proposed SDLs in the literature can be kernelized in a straightforward way.

A novel compression-based dissimilarity measure, particularly designed for textures, is also proposed. It is shown how it can be incorporated into the proposed kernelized SDL as a data-dependent kernel to significantly improve the accuracy of a pixel-based texture classification systems on benchmark datasets, such as the Brodatz album.

Eventually, the proposed SDL is extended to multiview representations. There are situations where there are more than one view/representation for a dataset. An effective way is proposed to make use of the complementary information available in all these representations, by learning one dictionary per view and computing the corresponding sparse coefficients. By fusing these coefficients, a multiview representation is provided where classification can be performed faster and more accurately. The effectiveness of this multiview SDL in emotion recognition applications will be also shown .

1.4 Organization

The organization of the rest of the thesis is as follows: Chapter 2 provides an overview on dictionary learning and sparse representation (DLSR). It first provides the formulation for

unsupervised dictionary learning, then extensively reviews many of the current supervised dictionary learning approaches in the literature and their shortcomings.

Chapter 3 provides the mathematical formulation for the proposed supervised dictionary learning approach. To this end, it first reviews the mathematical background for the proposed SDL, i.e., Hilbert Schmidt independence criterion (HSIC). Then provides the formulation for the proposed SDL and its kernelized version.

Chapter 4 presents the experimental setup and results on various datasets and in different applications such as face recognition, digit recognition, and texture classification.

The proposed compression-based dissimilarity measure and its properties are described in Chapter 5. This chapter first reviews the normalized information distance (NID) and its computable version, i.e., normalized compression distance (NCD). Then the formulation for the proposed measure is provided. Finally, it shows how by incorporating the proposed measure into the kernelized version of SDL, the performance of a texture classification system can be significantly improved.

Chapter 6 extends the proposed SDL to multiview and regression problems. The former is useful in applications where data is represented using more than one feature set, whereas the latter is needed when the information category is defined in continuous domain rather than a discrete one. The chapter shows the effectiveness of the proposed extensions to emotion recognition applications using speech and visual expressions. Finally, Chapter 7 concludes the thesis.

Chapter 2

Literature Review

In this chapter, an overview of the dictionary learning and sparse representation is provided. Also a brief review of recent attempts to make the approach more suitable for classification tasks is presented.

2.1 Unsupervised Dictionary Learning

Considering a finite training set of signals $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in \mathbb{R}^{p \times n}$, where p is the dimensionality and n is the number of data samples, according to classical dictionary learning and sparse representation (DLSR) techniques (refer to [1,13,14] for a recent review on this topic), these signals can be represented by a linear decomposition over a few dictionary atoms by minimizing a loss function as given below

$$L(\mathbf{X}, \mathbf{D}, \boldsymbol{\alpha}) = \sum_{i=1}^{n} l(\mathbf{x}_i, \mathbf{D}, \boldsymbol{\alpha}), \tag{2.1}$$

where $\mathbf{D} \in \mathbb{R}^{p \times k}$ is the dictionary of k atoms, and $\boldsymbol{\alpha} \in \mathbb{R}^{k \times n}$ are the coefficients.

This loss function can be defined in various ways based on the application in hand. However, what is common in DLSR literature is to define the loss function L as the

reconstruction error in a mean-squared sense, with a sparsity-inducing function ψ as a regularization penalty to ensure the sparsity of coefficients. Hence, (2.1) can be written as

$$L(\mathbf{X}, \mathbf{D}, \boldsymbol{\alpha}) = \min_{\mathbf{D},\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{X} - \mathbf{D}\boldsymbol{\alpha}\|_{\mathrm{F}}^2 + \lambda\psi(\boldsymbol{\alpha}), \tag{2.2}$$

where subscript F indicates the Frobenius norm and λ is the regularization parameter that affects the number of nonzero coefficients.

An intuitive measure of sparsity is ℓ_0-norm, which indicates the number of nonzero elements in a vector[1]. However, the optimization problem obtained from replacing sparsity-inducing function ψ in (2.2) with ℓ_0 is nonconvex, and the problem is NP-hard (refer to [14] for a recent comprehensive discussion on this issue). There are two main proposed approximate solutions to overcome this problem: the first is based on greedy algorithms, such as the well-known orthogonal matching pursuit (OMP) [11, 12, 14]; the second works by approximating a highly discontinuous ℓ_0-norm by a continuous function such as the ℓ_1-norm. This leads to an approach which is widely known in the literature as *lasso* [5] or *basis pursuit* (BP) [10], and (2.2) converts to

$$L(\mathbf{X}, \mathbf{D}, \boldsymbol{\alpha}) = \min_{\mathbf{D},\boldsymbol{\alpha}} \sum_{i=1}^{n} \left(\tfrac{1}{2}\|\mathbf{X}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{\alpha}_i\|_1 \right). \tag{2.3}$$

where $\boldsymbol{\alpha}_i$ is the i^{th} column of $\boldsymbol{\alpha}$.

In (2.3), the main optimization goal for the computation of the dictionary and sparse coefficients is minimizing the reconstruction error in the mean-squared sense. While this works well in applications where the primary goal is to reconstruct signals as accurately as possible, such as in denoising, image inpainting, and coding, it is not the ultimate goal in classification tasks [36], as discriminating signals is more important here. Hence, recently, there have been several attempts to include category information in computing either dictionary, coefficients, or both. In the following section, a brief overview of proposed supervised dictionary learning approaches in the literature will be provided. To this end, the proposed approaches are categorized into six different categories, while it is admitted that this taxonomy of approaches is not unique and could be done differently.

[1] ℓ_0 norm of vector $\mathbf{x}$ is defined as $\|\mathbf{x}\|_0 = \#\{i : x_i \neq 0\}$.

2.2 Supervised Dictionary Learning

As mentioned in the previous section, (2.3) provides a reconstructive formulation for computing the dictionary and sparse coefficients, given a set of data samples. Although the problem is not convex on both dictionary $\mathbf{D}$ and coefficients $\boldsymbol{\alpha}$, this optimization problem is convex if it is solved iteratively and alternately on these two unknowns. Several fast algorithms have recently been proposed for this purpose, such as K-SVD [37], online learning [38], and cyclic coordinate descent [39]. However, none of these approaches takes into account the category information for learning either the dictionary or the coefficients.

2.2.1 Learning One Dictionary per Class

The first and simplest approach to include category information in DLSR is computing one dictionary per class, i.e., using the training samples in each class to compute part of the dictionary, and then composing all these partial dictionaries into one. In providing the mathematical formulation for all the approaches in this category of SDL, it is always assumed that the training samples are grouped based on the classes they belong to such that $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_c] \in \mathbb{R}^{p \times n}$, where c is the number of classes and $\mathbf{X}_i = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, ..., \mathbf{x}_{im}] \in \mathbb{R}^{p \times m}$ is the group of training samples in class i.

Supervised k-means

Perhaps the earliest work in this direction is the one based on the so-called texton-based approach [19, 40–44]. The texton-based approach, can be considered a dictionary learning approach particularly tailored for texture analysis. In this approach, textons, which are computed using the k-means clustering algorithm over patches extracted from texture images, play the role of dictionary atoms. Although in a texton-based approach the texture images are usually modeled with a histogram of textons and hence, the approach falls mainly into the category of supervised dictionary learning explained in Subsection 2.2.6, the idea of using k-means and the computed cluster centers as the dictionary elements can still be considered here as a SDL approach that computes one dictionary per class.

Therefore, a specific name is suggested for this technique, i.e., supervised k-means, to differentiate it from a texton-based approach. In supervised k-means, k-means is applied to the training samples in each class, and the k cluster centers computed are considered to be the dictionary for this class. These partial dictionaries are eventually composed into one dictionary.

In the mathematical framework, each subdictionary $\mathbf{D}_i = [\mathbf{d}_{i1}, \mathbf{d}_{i2}, ..., \mathbf{d}_{ik_i}] \in \mathbb{R}^{p \times k_i}$ can be computed using the training samples in class i, i.e., using $\mathbf{X}_i = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, ..., \mathbf{x}_{im}] \in \mathbb{R}^{p \times m}$ and the optimization problem

$$\arg\min_{\mathbf{D}_i} \sum_{l=1}^{k_i} \sum_{\mathbf{x}_{ij} \in S_l} \|\mathbf{x}_{ij} - \mathbf{d}_{il}\| \tag{2.4}$$

where $S = \{S_1, S_2, ..., S_{k_i}\}$ are k_i clusters that partition data samples $\mathbf{X}_i$ in class i. Usually, k_i, the number of dictionary atoms computed per class, is the same over all classes. By composing all $\mathbf{D}_i$ into one dictionary such that $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, ..., \mathbf{D}_c] \in \mathbb{R}^{p \times k}$, where $k = k_i \cdot c$, the whole dictionary is obtained.

One can explain why it might be expected that a supervised k-means performs better than an unsupervised one by understanding how k-means compute the cluster centers: it essentially computes the cluster centers by taking the mean of the points. Hence, if k-means was applied to the data points across classes, the resultant cluster centers might not be corresponding to the data points in any of the classes, and consequently the resultant cluster centers would not be identified uniquely with individual classes. In other words, the cluster centers computed using k-means across classes would not be representing data samples in a class properly. Thus, in classification tasks, it will be beneficial, particularly at small dictionary sizes, to use k-means for the data points in one class at a time.

Sparse representation-based classification (SRC)

In [21], the training samples are used as the dictionary in face recognition and hence, this technique, called sparse representation-based classification (SRC), effectively falls into the same category as training one dictionary per class. However, no actual training is performed here, and the whole training samples are used directly in the dictionary.

To describe SRC more formally, suppose that $\mathbf{x}_{\text{ts}} \in \mathbb{R}^p$ is a test sample. The SRC algorithm assigns the whole training set $\mathbf{X}$ to the dictionary $\mathbf{D}$, and computes the sparse coefficients $\boldsymbol{\alpha}$ for test sample $\mathbf{x}_{\text{ts}}$ using the *lasso* given in (2.3) as follows

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{x}_{\text{ts}} - \mathbf{X}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1. \tag{2.5}$$

In the next step, the residual error is computed for the reconstruction of the test sample using training samples of each class and their corresponding sparse coefficients

$$r_i(\mathbf{x}_{\text{ts}}) = \|\mathbf{x}_{\text{ts}} - \mathbf{X}\delta_i(\boldsymbol{\alpha})\|_2^2, \tag{2.6}$$

where δ_i is a characteristic function that selects the coefficients associated with class i. This residual error is found for each class separately, and then the class label of the given test samples is assigned according to

$$label(\mathbf{x}_{\text{ts}}) = \arg\min_i \ r_i(\mathbf{x}_{\text{ts}}). \tag{2.7}$$

However, using the training samples as the dictionary in this approach results in a very large and possibly inefficient dictionary, due to the noisy training instances.

Metaface

To obtain a smaller dictionary, Yang *et al.* proposed an approach called *metaface*, which learns a smaller dictionary for each class and then composes them into one dictionary [45]. Metaface was originally proposed for the application of face recognition, but it is general and can be used in any application. In this approach, each subdictionary $\mathbf{D}_i$ is computed using the training samples $\mathbf{X}_i$ in class i using the *lasso*[2]

$$\min_{\mathbf{D}_i, \boldsymbol{\alpha}_i} \frac{1}{2}\|\mathbf{X}_i - \mathbf{D}_i\boldsymbol{\alpha}_i\|_{\text{F}}^2 + \lambda\|\boldsymbol{\alpha}_i\|_1. \tag{2.8}$$

[2]In this chapter, whenever ℓ_1-norm is used over a matrix, it is meant that ℓ_1-norms over each column of the matrix are summed such as what is used in (2.3). Hence the correct form for (2.8) is: $\min_{\mathbf{D},\boldsymbol{\alpha}} \sum_{j=1}^{m} \left(\frac{1}{2}\|\mathbf{X}_{ij} - \mathbf{D}\boldsymbol{\alpha}_{ij}\|_{\text{F}}^2 + \lambda\|\boldsymbol{\alpha}_{ij}\|_1\right)$. However, similar forms as in (2.8) are loosely used for ℓ_1-norm in the rest of this chapter to avoid too long and complex formulations and to focus more on the concept.

Since this optimization problem is nonconvex when both dictionary and coefficients are unknown, it has to be solved iteratively and alternately with one unknown variable considered fixed in each alteration. Computed subdictionaries are eventually composed into one dictionary $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, ..., \mathbf{D}_c] \in \mathbb{R}^{p \times k}$. After computation of the dictionary, the class label of a test sample $\mathbf{x}_{\text{ts}}$ is computed in the same way as explained in the SRC approach, i.e., by finding the coefficients for this test sample using the computed dictionary instead of the whole training set in (2.5), followed by the computation of the residuals given in (2.6), and assigning the test sample to the class that yields the minimal residue.

Although the metaface approach can potentially reduce the size of the dictionary compared to the SRC, its major drawback is that the training samples in one class are used for computing the atoms in the corresponding subdictionary, irrespective of the training samples from other classes. This means that if training samples across classes have some common properties, these shared properties cannot be learned in common in the dictionary.

Dictionary learning with structured incoherence (DLSI)

Ramirez *et al.* proposed to overcome the aforementioned problem with the metaface approach by including an incoherence term in (2.3) to encourage independency of dictionaries from different classes, while still allowing for different classes to share features [46].

To enable sharing features among the data points in different classes for learning the dictionary, instead of learning each $\mathbf{D}_i$ independently and unaware of data points in other classes, a coherence term is added to the *lasso* as described by the formulation below

$$\min_{\{\mathbf{D}_i, \boldsymbol{\alpha}_i\}_{i=1,...,c}} \sum_{i=1}^{c} \left\{ \|\mathbf{X}_i - \mathbf{D}_i \boldsymbol{\alpha}_i\|_{\text{F}}^2 + \lambda \|\boldsymbol{\alpha}_i\|_1 \right\} + \eta \sum_{i \neq j} \left\| \mathbf{D}_i^\top \mathbf{D}_j \right\|_{\text{F}}^2, \tag{2.9}$$

where the last term is an incoherence term $\mathcal{Q}(\mathbf{D}_i, \mathbf{D}_j)$, which has been proposed in [46] to be defined as the inner product between two subdictionaries $\mathbf{D}_i$ and $\mathbf{D}_j$ but it can be defined differently. After finding the dictionary, the classification of a test sample is performed the same way as with the SRC.

The main drawback of all the approaches in this first category of SDL is that they may

lead to a very large dictionary, as the size of the composed dictionary grows linearly with the number of classes.

2.2.2 Pruning Large Dictionaries

The second category of SDL approaches learn a very large dictionary unsupervised in the beginning, then merge the atoms in the dictionary by optimizing an objective function that takes into account the category information.

Information bottleneck (IB)

One major work in the literature in this direction is based on agglomerative information bottleneck (AIB), which iteratively merges two dictionary atoms that cause the smallest decrease in the mutual information between the dictionary atoms and the class labels [47]. The discriminative power of a dictionary $\mathbf{D}$ is characterized by the AIB as the amount of mutual information $I(d, y)$ shared by random variables d (dictionary atoms) and y (category information):

$$I(d, y) = \sum_{d \in \mathbf{D}} \sum_{y=1}^{c} P(d, y) log \frac{P(d, y)}{P(d)P(y)} \tag{2.10}$$

where the joint probability $P(d, y)$ is estimated from the data by counting the number of occurrences of dictionary atoms d in each category $y = \{1, ..., c\}$. The mutual information $I(d, y)$ is monotonically decreased as the AIB iteratively compresses the dictionary by merging dictionary atoms. This is continued until a predefined dictionary size is obtained. Although the approach is slow, a solution is proposed in [47] to make it computationally efficient.

Universal visual dictionary (UVD)

Another major work is based on merging two dictionary atoms so as to minimize the loss of mutual information between the histogram of dictionary atoms over signal constituents,

e.g., image patches, and class labels [48]. From this point of view, the difference between this approach and the one based on AIB is in the way they measure the discriminative power of the dictionary. In this approach, rather than measuring the discriminative power of the dictionary on individual dictionary atoms, it is measured on the histogram of dictionary atoms over signal constituents $\mathbf{H}$. Therefore, $I(h, y)$, where h is the random variable over the histograms $\mathbf{H}$ is considered in UVD, instead of $I(d, y)$ used by AIB. However, since the dimensionality of histograms tends to be very high, estimation of $I(h, y)$ is only possible with strong assumptions on the histograms. In [48], it is assumed that histograms can be modeled using a mixture of Gaussians, with one Gaussian per category. Based on this assumption, in [48], category posterior probability $p(y|h)$ is used instead of mutual information $I(h, y)$ for characterizing the discriminative power of the dictionary. Since this approach works on a histogram of dictionary atoms over signal constituents, it can be also categorized in the sixth category of SDL explained in Subsection 2.2.6.

One main drawback of this category of SDL is that the reduced dictionary obtained performs, at best, as well as the original one. Since the initial dictionary is learned unsupervised, even though with its large size it includes almost all possible atoms that helps to improve the performance of the classification task, the consecutive pruning stage is inefficient in terms of computational load. This can be significantly improved by finding a discriminative dictionary from the beginning.

2.2.3 Learning Dictionary and Classifier in One Optimization Problem

The third category of SDL, which is based on several research works published in [16, 49–53] can be considered a major leap in the field. In this category, the classifier parameters and the dictionary are learned in a joint optimization problem. Although this idea is more sophisticated than the previous two, its major disadvantage is that the optimization problem is nonconvex and complex. If it is done alternatively between learning the dictionary and classifier parameters, it is quite likely to become stuck in local minima. On the other hand, due to the complexity of the problem (except for the bilinear classifier in [16]), other papers only consider linear classifiers, which are usually too simple to solve difficult problems, and

can only be successful in simple classification tasks as shown in [16].

Supervised dictionary learning-discriminative (SDL-D)

Mairal *et al.* were one of the first research teams who proposed a joint optimization problem for learning the dictionary and the classifier parameters [16, 49, 53]. In [16] they proposed the following formulation

$$\min_{\mathbf{D},\mathbf{W},\boldsymbol{\alpha}} \left(\sum_{i=1}^{n} \mathcal{C}(y_i f(\mathbf{x}_i, \boldsymbol{\alpha}_i, \mathbf{W})) + \lambda_0 \left\| \mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i \right\|_2^2 + \lambda_1 \left\| \boldsymbol{\alpha}_i \right\|_1 \right) + \lambda_2 \left\| \mathbf{W} \right\|_2^2, \tag{2.11}$$

where $\mathcal{C}(x) = log(1 + e^{-x})$ is the logistic loss function, $(y_i \in \{-1, +1\})_{i=1}^{n}$ are binary class labels, $f(.)$ is the classifier function, and $\mathbf{W}$ is the associated classifier parameters to be learned. In (2.11), λ_0 is the parameter that controls the relative importance of the reconstruction error and the loss function on the classifier, λ_1 is the regularization parameter that controls the level of sparsity of the coefficients, and λ_2 is the regularization parameter to prevent overfitting the classifier. The actual discriminative formulation proposed in [16] is sufficiently more complex than (2.11) and its description is not provided here. The optimization problem in (2.11), is a nonconvex problem and has many parameters to tune, which makes the approach computationally expensive.

Discriminative K-SVD (DK-SVD)

In [50], Zhang and Li propose a technique called discriminative K-SVD (DK-SVD). DK-SVD truly jointly learns the classifier parameters and dictionary, without alternating between these two steps. This prevents the possibility of getting stuck in local minima. However, only linear classifiers are considered in DK-SVD, which may lead to poor performance in difficult classification tasks.

To provide the formulation for DK-SVD, one may notice that after learning the dictionary using the *lasso* (2.3), a linear classifier is to be learned on the coefficients $\boldsymbol{\alpha}$. Suppose that $\mathbf{W} \in \mathbb{R}^{c \times k}$ are the classifier parameters (c is the total number of classes and k is the number of dictionary atoms), and $\mathbf{H} \in \mathbb{R}^{c \times n}$ includes the class labels (n is the number of training samples) such that each column of $\mathbf{H}$ is $\mathbf{h}_i = \{0, ..., 1, ..., 0\}^\top$, i.e., there is exactly

one nonzero element in each column of $\mathbf{H}$, whose position indicates the class of the corresponding training sample. The classifier can be learnt using least square formulation by minimizing the classifier error in the mean-squared sense using the optimization problem

$$\min_{\mathbf{W}} \frac{1}{2}\|\mathbf{H} - \mathbf{W}\boldsymbol{\alpha}\|_{\mathrm{F}}^2. \tag{2.12}$$

This optimization problem can be combined with the *lasso* (2.3) into one optimization problem

$$\min_{\mathbf{D},\mathbf{W},\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{X} - \mathbf{D}\boldsymbol{\alpha}\|_{\mathrm{F}}^2 + \frac{\gamma}{2}\|\mathbf{H} - \mathbf{W}\boldsymbol{\alpha}\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{\alpha}\|_1. \tag{2.13}$$

To find the dictionary, coefficients, and the classifier, the optimization problem given in (2.13) has to be solved iteratively and alternately, with two of these unknowns fixed each time and solving for the third. This makes the solution very slow and very likely to get stuck in local minima. To partially overcome these problems, it is proposed in [50] to combine the first two terms in (2.13) into one term as follows

$$\min_{\mathbf{D},\mathbf{W},\boldsymbol{\alpha}} \frac{1}{2}\left\|\begin{bmatrix}\mathbf{X}\\ \sqrt{\gamma}\,\mathbf{H}\end{bmatrix} - \begin{bmatrix}\mathbf{D}\\ \sqrt{\gamma}\,\mathbf{W}\end{bmatrix}\boldsymbol{\alpha}\right\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{\alpha}\|_1. \tag{2.14}$$

Considering $\begin{bmatrix}\mathbf{X}\\ \sqrt{\gamma}\,\mathbf{H}\end{bmatrix}$ as a new training set $\mathbf{X}_{\mathrm{N}} \in \mathbb{R}^{(p+c)\times n}$ and $\begin{bmatrix}\mathbf{D}\\ \sqrt{\gamma}\,\mathbf{W}\end{bmatrix}$ as a new dictionary $\mathbf{D}_{\mathrm{N}} \in \mathbb{R}^{(p+c)\times k}$, (2.14) is converted to the *lasso*

$$\min_{\mathbf{D}_{\mathrm{N}},\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{X}_{\mathrm{N}} - \mathbf{D}_{\mathrm{N}}\boldsymbol{\alpha}\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{\alpha}\|_1, \tag{2.15}$$

and can be efficiently solved by one of the recently developed fast algorithms for this purpose such as K-SVD [37]. Deriving $\mathbf{D}$ and $\mathbf{W}$ from $\mathbf{D}_{\mathrm{N}}$ is straightforward and the details are provided in [50].

One major problem with the approaches in this category of SDL is that there exist many parameters involved in the formulation, which are hard and time-consuming to tune (see for example [16, 53]).

2.2.4 Including Category Information in the Learning of the Dictionary

The fourth category of SDL approaches includes the category information in the learning of the dictionary.

Information loss minimization (info-loss)

In [54], it is proposed to include category information into the learning of the dictionary, by minimizing the information loss due to predicting labels from a supervised dictionary learned instead of original training data samples. This approach is known as *info-loss* in the SDL literature. In fact, in supervised dictionary learning, the ultimate goal is to represent the original high-dimensional feature space by a dictionary such that it can facilitate the prediction of the class labels correctly. Ideally, the dictionary should maintain all discriminative power of the original feature space. However, some of this information is lost during the quantization of the feature space. In [54], it is proposed to learn the dictionary such that the information loss

$$I(x, y) - I(d, y) \tag{2.16}$$

is minimized, where I indicates the mutual information between its arguments as random variables, and x, d, and y are the random variables on the original feature space $\mathbf{X}$, learned dictionary $\mathbf{D}$, and the class labels $\mathbf{Y}$, respectively.

Just the same as in the previous category of SDL, the info-loss approach has the major drawback that it may become stuck in local minima. This is mainly because the optimization has to be done iteratively and alternately on two updates, as there is no closed-form solution for the approach (the details of the approach have not been provided here; interested reader can refer to the original paper for more information).

Randomized clustering forests (RCF)

In [55], it is proposed to learn the dictionary atoms using extremely randomized decision trees. This approach can also fall into the second category of SDLs, as it seems that it

starts from a very large dictionary using random forests, and tries to prune it later to conclude with a smaller dictionary.

2.2.5 Including Category Information in the Learning of the Sparse Coefficients

The fifth category of SDL includes class category in the learning of coefficients [36] or in the learning of both dictionary and coefficients [22, 56]. Supervised coefficient learning in all these papers [22, 36, 56] has been performed more or less in the same way using the Fisher discrimination criterion [57], i.e., by minimizing the within-class covariance of coefficients and at the same time maximizing their between-class covariance. As for the dictionary, while [36] uses predefined bases by deploying an overcomplete dictionary as a combination of Haar and Gabor bases, [22] proposes a discriminative fidelity term to learn the dictionary, for which further description is provided below, along with the learning of the coefficients.

Fisher discrimination dictionary learning (FDDL)

In [22], an approach called Fisher discrimination dictionary learning (FDDL) is proposed, that uses category information in learning both dictionary and sparse coefficients. To learn the dictionary supervised, a discriminative fidelity term is proposed that encourages learning dictionary atoms of one class from the training samples of the same class, and at the same time penalizes their learning by the training samples from other classes. As stated above, the coefficients are learned supervised, by including the Fisher discriminant criterion in their learning.

To provide a mathematical formulation for FDDL, suppose that the training samples are grouped according to the classes they belong to, i.e., $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_c] \in \mathbb{R}^{p \times n}$, where c is the number of classes. The objective function in FDDL consists of two terms: a fidelity term and a discrimination constraint term on coefficients

$$J_{(\mathbf{D},\boldsymbol{\alpha})} = \min_{\mathbf{D},\boldsymbol{\alpha}} \left\{ r(\mathbf{X}, \mathbf{D}, \boldsymbol{\alpha}) + \lambda_1 \|\boldsymbol{\alpha}\|_1 + \lambda_2 f(\boldsymbol{\alpha}) \right\}, \tag{2.17}$$

where $r(\mathbf{X}, \mathbf{D}, \boldsymbol{\alpha})$ is the fidelity term and $f(\boldsymbol{\alpha})$ is the discrimination constraint on the coefficients.

The fidelity term is defined in [22] as follows

$$r(\mathbf{X}, \mathbf{D}, \boldsymbol{\alpha}) = \|\mathbf{X}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_{\mathrm{F}}^2 + \left\|\mathbf{X}_i - \mathbf{D}_i\boldsymbol{\alpha}_i^i\right\|_{\mathrm{F}}^2 + \sum_{\substack{j=1 \\ j\neq i}}^{c} \left\|\mathbf{D}_j\boldsymbol{\alpha}_i^j\right\|_{\mathrm{F}}^2, \tag{2.18}$$

where $\mathbf{D}_i$ is the part of the dictionary associated with class i, and $\boldsymbol{\alpha}_i$ is the representation of $\mathbf{X}_i$ over $\mathbf{D}$. Also $\boldsymbol{\alpha}_i = [\boldsymbol{\alpha}_i^1, \boldsymbol{\alpha}_i^2, ..., \boldsymbol{\alpha}_i^c]$, where $\boldsymbol{\alpha}_i^j$ is the part of the coefficients that represent $\mathbf{X}_i$ over the subdictionary $\mathbf{D}_j$. In (2.18), the first two terms indicate that the whole dictionary and also the subdictionary associated with class i should well represent the data samples in the same class $\mathbf{X}_i$, whereas the last term indicates that the subdictionaries from other classes have little contribution towards the representation of the data samples in class i.

The Fisher discrimination term, on the other hand, is as follows

$$f(\boldsymbol{\alpha}) = \mathrm{tr}(S_{\mathrm{W}}(\boldsymbol{\alpha})) - \mathrm{tr}(S_{\mathrm{B}}(\boldsymbol{\alpha})) + \eta \|\boldsymbol{\alpha}\|_{\mathrm{F}}^2, \tag{2.19}$$

where tr is the trace operator; S_{W} and S_{B} are within- and between-class covariance matrices, respectively. The last term is a penalty added to (2.19) to make the optimization problem convex [22].

The joint optimization problem, due to the Fisher discrimination criterion on the coefficients and the discriminative fidelity term on the dictionary proposed in (2.17), is not convex, and has to be solved alternately and iteratively between these two terms until it converges. However, there is no guarantee to find the global minimum. Also, it is not clear whether the improvement obtained in classification by including the Fisher discriminant criterion on coefficients justifies the additional computation load imposed on the learning, as there is no comparison provided in [22] on the classification with and without including supervision on coefficients.

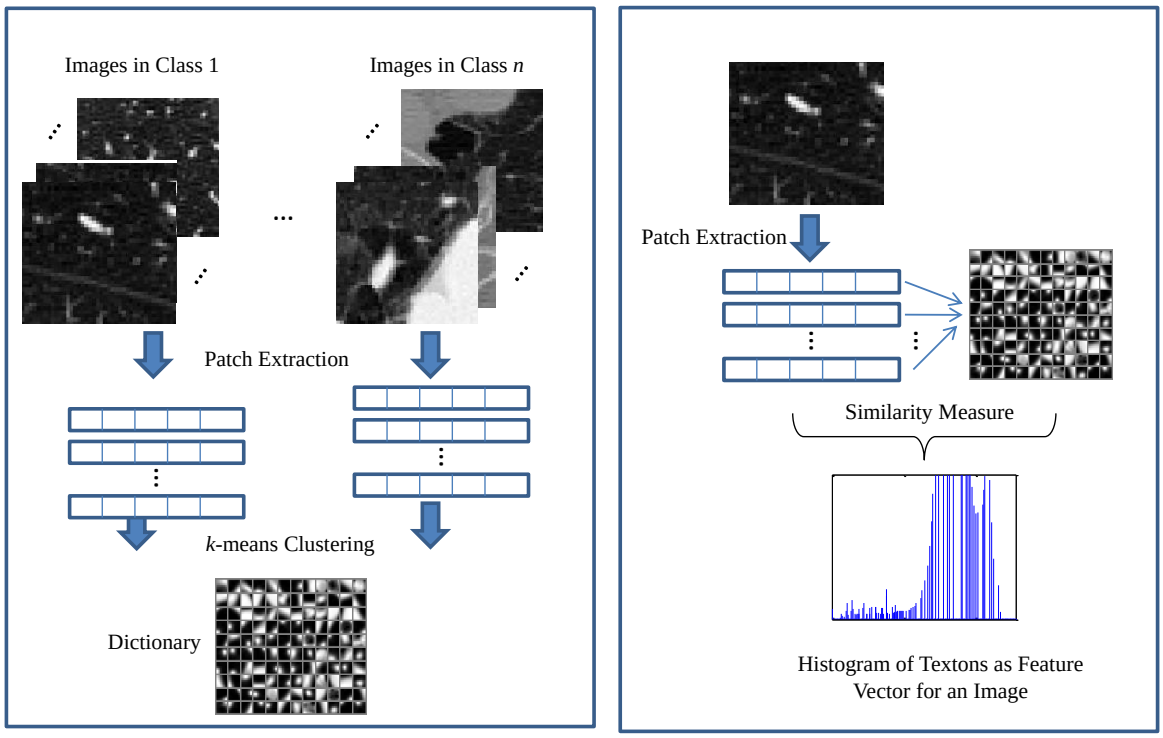
2.2.6 Learning a Histogram of Dictionary Elements over the Signal Constituents

There are situations where a signal is made of some local constituents, e.g., an image is made up of patches. However, the ultimate classification task is to classify the signal, not its individual local constituents, e.g., the whole image, not the patches in the previous example. This classification task is usually tackled by computing the histogram of dictionary atoms computed over local constituents of a signal. The computed histograms are used as the signature (model) of the signal, which are eventually used for the training of a classifier and predicting the labels of unknown signals. Unlike the previous five categories, the motivation of the approaches in the sixth SDL category is to design a supervised dictionary which is discriminative over the histogram representation of signals, not over individual local descriptors [58–60]. Hence, these approaches cannot be used in cases where a signal does not consist of a collection of local constituents.

Texton-based approach

The texton-based approach [19,40–44], is one of the earliest that was proposed to compute the histogram of dictionary elements, called textons, to model a texture image based on patches extracted. This approach was particularly proposed for texture analysis, but is sufficiently general to be used in other applications. In a texton-based approach, the first step is to construct the dictionary. To this end, small-sized local patches are randomly extracted from each texture image in the training set. These small patches are then aggregated over all images in a class, and clustered using a clustering algorithm such as k-means. Obtained cluster centers form a dictionary that represents the class of textures used. In other words, supervised k-means is used to compute the dictionary atoms [19,44].

The next step is to find the features (learn the model) using the images in the training set. To this end, small patches of the same size as the previous step are extracted by sliding a window over each training image in a class. Then the distance between each patch to all textons in the dictionary are computed, to find the closest match using a distance measure such as Euclidean distance. Finally, a histogram of textons is updated accordingly for



Construction of Dictionary (a)

Learning Model (b)

Figure 2.1: The illustration of two steps of a texton-based system: (a) the generation of texton dictionary using supervised k-means (b) and the generation of features by computing the texton histograms on an image (from [25]).

each image based on the closest match found. This yields a histogram for each image in the training set, which is used as the features representing that image after normalization. Figure 2.1 illustrates the construction of the dictionary and learning of the model in a texton-based system.

Histogram computation using dictionary learning and sparse representation

In the texton-based approach, supervised k-means was used to compute the dictionary. To compute the histogram of textons, each patch was represented by the closest match in the dictionary. This is the maximum sparsity possible as each patch is represented by only one dictionary element. However, as proposed in [17], it is possible to use (2.3) and one of the recent algorithms for its implementation, such as online learning [38], to compute the dictionary and the corresponding sparse coefficients over the patches extracted from an image. The same as the texton-based approach, building the dictionary and histogram of dictionary elements can be done in two steps. In the first step, random patches are extracted from each image in the training set. Next, by submitting these patches into the online learning algorithm, the dictionary can be computed [17].

As the second step, it is needed to find the model (feature set) for each image. To this end, patches of the same size as those in the dictionary learning step are extracted from each image, i.e., $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in \mathbb{R}^{p \times n}$, where n is the number of patches extracted, and each patch size is $\sqrt{p} \times \sqrt{p}$. Then using (2.3), the corresponding coefficients $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_n] \in \mathbb{R}^{k \times n}$ are computed. For each patch $\mathbf{x}_i$, most of the elements in the corresponding coefficient $\boldsymbol{\alpha}_i$ are zero. The nonzero elements in $\boldsymbol{\alpha}_i$ determine the atoms in the dictionary $\mathbf{D}$ that contribute towards the representation of the patch $\boldsymbol{\alpha}_i$. If all these coefficients are summed up for all patches extracted from an image, one can effectively find the histogram of primitive elements contributing towards the representation of this particular image, i.e.,

$$\mathbf{H}(\mathbf{X}) = \sum_{i=1}^{n} \boldsymbol{\alpha}_i. \tag{2.20}$$

A histogram $\mathbf{H}$ with positive values in all bins can be eventually obtained by imposing a positive constraint on $\boldsymbol{\alpha}_i$ in (2.3). The positive constraint also prevents canceling the effect of different patches when they are summed up in (2.20). Equation (2.3) can be written as follows to consider this constraint as well

$$\begin{aligned} \min_{\mathbf{D},\boldsymbol{\alpha}} \quad & \sum_{i=1}^{n} \left(\tfrac{1}{2} \|\mathbf{X}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_{\mathrm{F}}^2 + \lambda \|\boldsymbol{\alpha}_i\|_1 \right), \\ \text{s.t.} \quad & \boldsymbol{\alpha}_i \geq 0 \quad \forall i = 1, ..., n \end{aligned} \tag{2.21}$$

In this way, while in a texton-based approach each patch is represented using only the closest texton in the dictionary, here each patch is represented by using several primitive elements in the dictionary, and hence can potentially provide richer representation than the texton-based approach. The number of nonzero elements in $\boldsymbol{\alpha}_i$ can be controlled using λ in (2.21), i.e., larger values of λ yield sparser coefficients [38].

Universal and adapted vocabularies (UAV)

The above two approaches do not include the class labels into the learning of the histograms. In [60], it is proposed to learn one bipartite histogram per class for each image. Each bipartite histogram, as the name implies, has two parts: a part adapted to the specific class, and a universal part. In each histogram, ideally, if the object belongs to the class, its adapted part is more significant than the universal one; otherwise the universal part is more dominant.

Gaussian mixture models (GMM) are used to learn the universal vocabularies (dictionaries) using maximum likelihood estimation (MLE) for low level local descriptors such as scale-invariant feature transform (SIFT) descriptors. Then class specific vocabularies are adapted by the maximum a posteriori (MAP) criterion. Eventually, the bipartite histograms are estimated by using the adapted and universal vocabularies [60].

Supervised dictionary learning model (SDLM)

A supervised dictionary learning model (SDLM) is proposed in [58], which combines an unsupervised model based on a Gaussian mixture model (GMM) with a supervised model, i.e., a logistic regression model in a probabilistic framework. As explained in the beginning of this subsection, the motivation of this model is to learn the dictionary such that the histogram representation of images are sufficiently discriminative over different classes. Intuitively, in SDLM, a logistic loss function is used to pass the discriminative information in class labels to histogram features. This information is subsequently passed to the dictionary learned over image local features by affecting the GMM parameters [58].

Figure 2.2 summarizes the taxonomy of dictionary learning and sparse representation techniques as presented in this chapter for quick reference.

In the next chapter, the mathematical formulation for the proposed approach will be explained, which I believe belongs to the fourth category of SDLs explained above, i.e., including category information to learn a supervised dictionary.

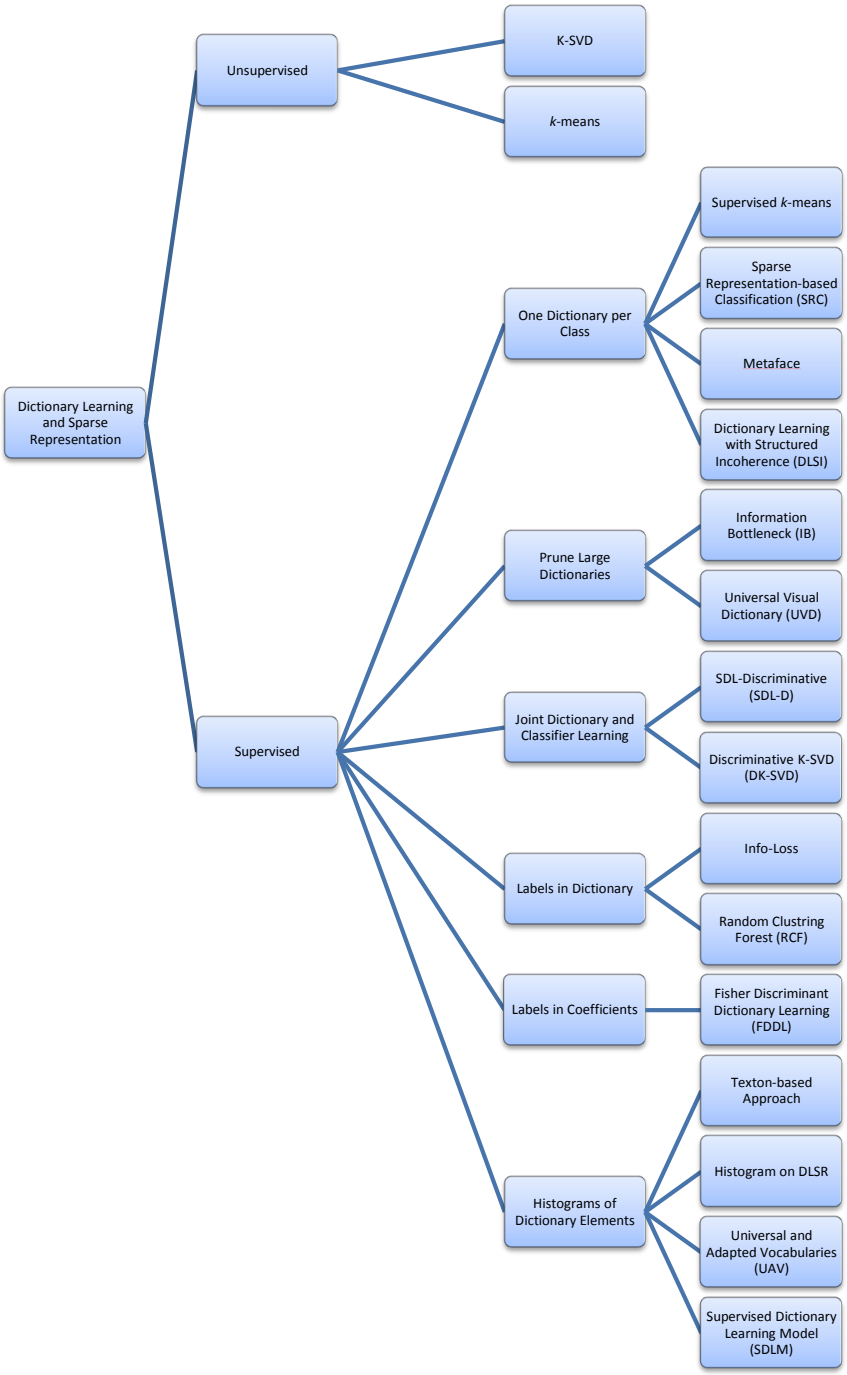


Figure 2.2: Taxonomy of dictionary learning and sparse representation as presented in this chapter. Supervised dictionary learning approaches are divided into six categories.

Chapter 3

Proposed Supervised Dictionary Learning

Given a finite set of data samples $\mathbf{X} \in \mathbb{R}^{p \times n}$, where p is the dimensionality of the data and n is the number of data samples, in this chapter, we address the problem of linear decomposition of $\mathbf{X}$ over a learned dictionary $\mathbf{D} \in \mathbb{R}^{d \times k}$, where k is the number of dictionary atoms, by minimizing a loss function. The loss function is defined as the reconstruction error and ℓ_1-norm is added as the regularization penalty. The goal is to make the dictionary $\mathbf{D}$ sufficiently discriminative over the classes.

To address this problem, we incorporate the class labels associated with the data samples into the learning of the dictionary to make it discriminative. To incorporate the class labels into dictionary learning, it is proposed to decompose signals using some learned bases that represent them in a space where the dependency between the signals and their corresponding class labels is maximized. To this end, a(n) (in)dependency test measure between two random variables is needed. Here, it is proposed to use the Hilbert-Schmidt independence criterion (HSIC) as the (in)dependency measure. In this chapter, we first describe HSIC, and then provide the formulation for the proposed supervised dictionary learning (SDL) approach. Subsequently, kernelized SDL is formulated to enable embedding kernels, including data-dependent ones, into the proposed SDL. This can significantly improve the discrimination power of the designed dictionary, which is essential in difficult

classification tasks, as will be shown in our experiments in Section 5.6 later.

3.1 Hilbert Schmidt Independence Criterion

There are several techniques in the literature to measure the (in)dependence of random variables including mutual information [61] and the Kullback-Leibler (KL) divergence [62]. In addition to these measures, there has recently been great interest in measuring (in)dependency using criteria based on functions in reproducing kernel Hilbert spaces (RKHSs). Bach and Jordan [63] were the first to accomplish this, by introducing kernel dependence functionals that significantly outperformed alternative approaches. Later, Gretton *et al.* [32] proposed another kernel-based approach called the Hilbert-Schmidt independence criterion (HSIC) to measure the (in)dependence of two random variables $\mathcal{X}$ and $\mathcal{Y}$. Since its introduction, the HSIC has been used in many applications, including feature selection [64], independent component analysis [65], and sorting/matching [66].

One can derive HSIC as a measure of (in)dependence between two random variables $\mathcal{X}$ and $\mathcal{Y}$ using two different approaches: first by computing the Hilbert-Schmidt norm of the cross-covariance operators in RKHSs as shown in [32, 33]; or second, by computing maximum mean discrepancy (MMD) of two distributions mapped to a high dimensional space, i.e., computed in RKHSs [67,68]. I believe that this latter approach is more straightforward and hence, use it to describe HSIC.

Let $\mathcal{Z} := \{(\mathbf{x}_1, \mathbf{y}_1,), ..., (\mathbf{x}_n, \mathbf{y}_n)\} \subseteq \mathcal{X} \times \mathcal{Y}$ be n independent observations drawn from $p := P_{\mathcal{X}\times\mathcal{Y}}$. To investigate whether $\mathcal{X}$ and $\mathcal{Y}$ are independent, one needs to determine whether distribution p factorizes, i.e., whether p is the same as $q := P_{\mathcal{X}} \times P_{\mathcal{Y}}$.

The means of the distributions are defined as follows

$$\mu[P_{\mathcal{X}\times\mathcal{Y}}] := \mathbf{E}_{xy}[v((x,y),.)], \tag{3.1}$$

$$\mu[P_{\mathcal{X}} \times P_{\mathcal{Y}}] := \mathbf{E}_x\mathbf{E}_y[v((x,y),.)], \tag{3.2}$$

where $\mathbf{E}_{xy}$ is the expectation over $(x, y) \sim P_{\mathcal{X}\times\mathcal{Y}}$ and kernel $v((x, y), (x', y'))$ is defined in RKHS over $\mathcal{X} \times \mathcal{Y}$. By computing the mean of distributions p and q in RKHS, higher

order statistics than the first order are effectively taken into account by mapping these distributions to a high-dimensional feature space. Hence, one can use $\text{MMD}(p,q) := \|\mu[P_{\mathcal{X}\times\mathcal{Y}}] - \mu[P_{\mathcal{X}} \times P_{\mathcal{Y}}]\|_2$ as a measure of (in)dependence of the random variables $\mathcal{X}$ and $\mathcal{Y}$. The higher the value of MMD, the closer the two distributions p and q and hence, the more dependent are random variables $\mathcal{X}$ and $\mathcal{Y}$.

Now suppose that $\mathcal{H}$ and $\mathcal{G}$ are two RKHSs in $\mathcal{X}$ and $\mathcal{Y}$, respectively. Hence, by the Riesz representation theorem, there are feature mappings $\phi(x) : \mathcal{X} \to \mathbb{R}$ and $\psi(y) : \mathcal{Y} \to \mathbb{R}$ such that $k(x,x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$ and $l(y,y') = \langle \psi(y), \psi(y') \rangle_{\mathcal{G}}$. Moreover, suppose that $v((x,y),(x',y')) = k(x,x')l(y,y')$, i.e., the RKHS is a direct product of $\mathcal{H} \otimes \mathcal{G}$ of the RKHSs on $\mathcal{X}$ and $\mathcal{Y}$. Then $\text{MMD}(p,q)$ can be written as

$$\begin{aligned} \text{MMD}^2(p,q) &= \|\mathbf{E}_{xy}[k(x,.)l(y,.)] \\ &\quad - \mathbf{E}_x[k(x,.)]\mathbf{E}_y[l(y,.)]\|_2^2 \\ &= \mathbf{E}_{xy}\mathbf{E}_{x'y'}[k(x,x')l(y,y')] \\ &\quad - 2\mathbf{E}_x\mathbf{E}_y\mathbf{E}_{x'y'}[k(x,x')l(y,y')] \\ &\quad + \mathbf{E}_x\mathbf{E}_y\mathbf{E}_{x'}\mathbf{E}_{y'}[k(x,x')l(y,y')]. \end{aligned} \tag{3.3}$$

This is exactly the HSIC, and equivalent to the Hilbert-Schmidt norm of the cross-covariance operator in RKHSs [32].

For practical purposes, HSIC has to be estimated using a finite number of data samples. Considering $\mathcal{Z} := \{(\mathbf{x}_1, \mathbf{y}_1,), ..., (\mathbf{x}_n, \mathbf{y}_n)\} \subseteq \mathcal{X} \times \mathcal{Y}$ as n independent observations drawn from $p := P_{\mathcal{X}\times\mathcal{Y}}$, an empirical estimate of HSIC is defined as follows [32]

$$\text{HSIC}(\mathcal{Z}) = \frac{1}{(n-1)^2}\text{tr}(\mathbf{KHLH}), \tag{3.4}$$

where tr is the trace operator, $\mathbf{H}, \mathbf{K}, \mathbf{L} \in \mathbb{R}^{n\times n}, K_{i,j} = k(x_i, x_j), L_{i,j} = l(y_i, y_j)$, and $\mathbf{H} = \mathbf{I} - n^{-1}\mathbf{e}\mathbf{e}^\top$ ($\mathbf{I}$ is the identity matrix, and $\mathbf{e}$ is a vector of n ones, and hence, $\mathbf{H}$ is the centering matrix [69]). It is important to notice that according to (3.4), to maximize the dependency between two random variables $\mathcal{X}$ and $\mathcal{Y}$, the empirical estimate of HSIC, i.e., $\text{tr}(\mathbf{KHLH})$ should be maximized.

3.2 Supervised Dictionary Learning Formulation

To formulate the proposed SDL, one can start from the reconstruction error given in (2.3). Let there be a finite training set of n data points, each of which consists of p features, i.e., $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in \mathbb{R}^{p \times n}$. Also suppose that features in data samples are centered, i.e., their mean is removed and hence, each row of $\mathbf{X}$ sums to zero. The problem statement is to find a linear decomposition of data $\mathbf{X} \in \mathbb{R}^{p \times n}$ using some bases $\mathbf{U} \in \mathbb{R}^{p \times k}$ such that the reconstruction error is minimum in the mean-squared sense, i.e.,

$$\min_{\mathbf{U}, V_i} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{U}\mathbf{v}_i\|_2^2 \,, \tag{3.5}$$

where $\mathbf{v}_i$ is the vector of k reconstruction coefficients. Equation (3.5) can be rewritten in matrix form as follows

$$\min_{\mathbf{U}, \mathbf{V}} \|\mathbf{X} - \mathbf{U}\mathbf{V}\|_{\mathrm{F}}^2 \,, \tag{3.6}$$

where $\mathbf{V} \in \mathbb{R}^{k \times n}$ is the matrix of coefficients. Since both $\mathbf{U}$ and $\mathbf{V}$ are unknown, this problem is ill-posed and does not have a unique solution unless some constraints are imposed on the bases $\mathbf{U}$. If one, for example, assumes that the columns of $\mathbf{U}$ are orthonormal, i.e., $\mathbf{U}^\top \mathbf{U} = \mathbf{I}$, (3.6) can be written as a constrained optimization problem as follows

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{V}} \quad & \|\mathbf{X} - \mathbf{U}\mathbf{V}\|_{\mathrm{F}}^2 \,. \\ \text{s.t.} \quad & \mathbf{U}^\top \mathbf{U} = \mathbf{I} \end{aligned} \tag{3.7}$$

To further investigate the optimization problem in (3.7), one can assume that the matrix $\mathbf{U}$ is fixed, and find the optimum matrix of coefficients $\mathbf{V}$ in terms of $\mathbf{X}$ and $\mathbf{U}$ by taking the derivative of the objective function given in (3.7) in respect to $\mathbf{V}$

$$\begin{aligned} \frac{\partial}{\partial \mathbf{V}} \|\mathbf{X} - \mathbf{U}\mathbf{V}\|_{\mathrm{F}}^2 &= \frac{\partial}{\partial \mathbf{V}} \mathrm{tr}[(\mathbf{X} - \mathbf{U}\mathbf{V})^\top (\mathbf{X} - \mathbf{U}\mathbf{V})] \\ &= \frac{\partial}{\partial \mathbf{V}} [\mathrm{tr}(\mathbf{X}^\top \mathbf{X}) - 2\mathrm{tr}(\mathbf{X}^\top \mathbf{U}\mathbf{V}) \\ &\quad + \mathrm{tr}(\mathbf{V}^\top \mathbf{U}^\top \mathbf{U}\mathbf{V})] \\ &= -2\mathbf{U}^\top \mathbf{X} + 2\mathbf{U}^\top \mathbf{U}\mathbf{V}. \end{aligned}$$

Equating the above derivative to zero and knowing that $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$, one obtains

$$\mathbf{V} = \mathbf{U}^\top\mathbf{X}. \tag{3.8}$$

By plugging the $\mathbf{V}$ found in (3.8) into the objective function of (3.7) we obtain

$$\begin{aligned}
\min_{\mathbf{U}} \left\|\mathbf{X} - \mathbf{U}\mathbf{U}^\top\mathbf{X}\right\|_{\mathrm{F}}^2 &= \min_{\mathbf{U}} \mathrm{tr}[(\mathbf{X} - \mathbf{U}\mathbf{U}^\top\mathbf{X})^\top(\mathbf{X} - \mathbf{U}\mathbf{U}^\top\mathbf{X})] \\
&= \min_{\mathbf{U}} [\mathrm{tr}(\mathbf{X}^\top\mathbf{X}) - 2\mathrm{tr}(\mathbf{X}^\top\mathbf{U}\mathbf{U}^\top\mathbf{X}) \\
&\quad + \mathrm{tr}(\mathbf{X}^\top\mathbf{U}\mathbf{U}^\top\mathbf{U}\mathbf{U}^\top\mathbf{X})] \\
&= \max_{\mathbf{U}} \mathrm{tr}(\mathbf{X}^\top\mathbf{U}\mathbf{U}^\top\mathbf{X}) \\
&= \max_{\mathbf{U}} \mathrm{tr}((\mathbf{U}^\top\mathbf{X})^\top\mathbf{U}^\top\mathbf{X}).
\end{aligned}$$

Let $\mathbf{K} = (\mathbf{U}^\top\mathbf{X})^\top\mathbf{U}^\top\mathbf{X}$, which is a linear kernel on the transformed data in the subspace $\mathbf{U}^\top\mathbf{X}$; recalling that the features are centered in the original space, multiplying the data $\mathbf{X}$ by the centering matrix $\mathbf{H}$ does not have any effect. Hence, one can write

$$\begin{aligned}
\max_{\mathbf{U}} \mathrm{tr}((\mathbf{U}^\top\mathbf{X})^\top\mathbf{U}^\top\mathbf{X}) &= \max_{\mathbf{U}} \mathrm{tr}((\mathbf{U}^\top\mathbf{X}\mathbf{H})^\top\mathbf{U}^\top\mathbf{X}\mathbf{H}\mathbf{I}) \\
&= \max_{\mathbf{U}} \mathrm{tr}(\mathbf{H}(\mathbf{U}^\top\mathbf{X})^\top\mathbf{U}^\top\mathbf{X}\mathbf{H}\mathbf{I}) \\
&= \max_{\mathbf{U}} \mathrm{tr}([(\mathbf{U}^\top\mathbf{X})^\top\mathbf{U}^\top\mathbf{X}]\mathbf{H}\mathbf{I}\mathbf{H}) \\
&= \max_{\mathbf{U}} \mathrm{tr}(\mathbf{K}\mathbf{H}\mathbf{I}\mathbf{H}),
\end{aligned} \tag{3.9}$$

where $\mathbf{I}$ is the identity matrix. To derive (3.9), the identities $\mathbf{H}^\top = \mathbf{H}$ and $\mathbf{X}\mathbf{H} = \mathbf{X}\mathbf{H}\mathbf{I}$ are used and it is also noted that the trace operator is invariant to the rotation of its arguments.

To enable providing an interpretation for (3.9), we recall that identity matrix $\mathbf{I}$ represents a kernel on a random variable, where each data sample has maximum similarity to itself and no similarity, whatsoever, to others. Hence, based on empirical HSIC, the objective function given in (3.9) indicates that the transformation $\mathbf{U}$ transforms the centered data[1] $\mathbf{X}\mathbf{H}$ to a space where the dependency of random variables x and another random

[1]Here, centered data means that the features are centered, not individual data samples.

variable whose kernel is identity matrix $\mathbf{I}$ is maximized. This means that using transformation $\mathbf{U}$, the random variable x is transformed such that each data sample has maximum similarity/correlation to itself and no similarity to other data samples. It is well known in the literature that these bases are the principal components of the signal $\mathbf{X}$ that represent the data in an uncorrelated space. With a few manipulations, the objective function given in (3.9) can be rewritten as follows:

$$\begin{aligned}\max_{\mathbf{U}} \operatorname{tr}((\mathbf{U}^\top\mathbf{X})^\top\mathbf{U}^\top\mathbf{X}) &= \max_{\mathbf{U}} \operatorname{tr}((\mathbf{U}^\top\mathbf{X}\mathbf{H})^\top\mathbf{U}^\top\mathbf{X}\mathbf{H}\mathbf{I}) \\ &= \max_{\mathbf{U}} \operatorname{tr}(\mathbf{H}\mathbf{X}^\top\mathbf{U}\mathbf{U}^\top\mathbf{X}\mathbf{H}\mathbf{I}) \\ &= \max_{\mathbf{U}} \operatorname{tr}(\mathbf{U}^\top\mathbf{X}\mathbf{H}\mathbf{I}\mathbf{H}\mathbf{X}^\top\mathbf{U}).\end{aligned}$$

In other words, it is shown that the optimization problem in (3.7) is equivalent to

$$\begin{aligned}&\max_{\mathbf{U}} \quad \operatorname{tr}(\mathbf{U}^\top\mathbf{X}\mathbf{H}\mathbf{I}\mathbf{H}\mathbf{X}^\top\mathbf{U}), \\ &\text{s.t.} \quad \mathbf{U}^\top\mathbf{U} = \mathbf{I}\end{aligned} \tag{3.10}$$

According to the Rayleigh-Ritz Theorem [70], the solution of the optimization problem in (3.10) is the top eigenvectors of $\mathbf{\Phi} = \mathbf{X}\mathbf{H}\mathbf{I}\mathbf{H}\mathbf{X}^\top$ corresponding to the largest eigenvalues of $\mathbf{\Phi}$. Here, $\mathbf{X}\mathbf{H}\mathbf{I}\mathbf{H}\mathbf{X}^\top$ is the covariance matrix of $\mathbf{X}$.

To summarize, it was shown above that the linear decomposition of signals that minimizes the reconstruction error in the mean-squared sense, represents the data in an uncorrelated space. This is, in fact, the same as in the principal component analysis (PCA), where the top eigenvectors of the covariance matrix are computed. However, as mentioned before, although minimization of reconstruction error is the ultimate goal in applications such as denoising and coding, in classification tasks, the main goal is maximum discrimination of classes. Hence, one is looking for a decomposition that represents the data in a space where the decomposed data have maximum dependency with their labels. To this end, we propose the new optimization problem as follows

$$\begin{aligned}&\max_{\mathbf{U}} \quad \operatorname{tr}(\mathbf{U}^\top\mathbf{X}\mathbf{H}\mathbf{L}\mathbf{H}\mathbf{X}^\top\mathbf{U}), \\ &\text{s.t.} \quad \mathbf{U}^\top\mathbf{U} = \mathbf{I}\end{aligned} \tag{3.11}$$

where $\mathbf{L}$ is a kernel, e.g., a linear kernel, on the labels $\mathbf{Y} \in \{0,1\}^{c\times n}$, i.e., $\mathbf{L} = \mathbf{Y}^\top\mathbf{Y}$ and c is the number of classes. Here, each column of $\mathbf{Y}$ is $\mathbf{y}_i = \{0,...,1,...,0\}^\top$. In other words,

Algorithm 1 Supervised Dictionary Learning

Input: Training data, $\mathbf{X}_{\text{tr}}$, test data, $\mathbf{X}_{\text{ts}}$, kernel matrix of labels $\mathbf{L}$, training data size, n, size of dictionary, k.

Output: Dictionary, $\mathbf{D}$, coefficients for training and test data, $\boldsymbol{\alpha}_{\text{tr}}$ and $\boldsymbol{\alpha}_{\text{ts}}$.

1: $\mathbf{H} \leftarrow \mathbf{I} - n^{-1}\mathbf{e}\mathbf{e}^{\top}$
2: $\boldsymbol{\Phi} \leftarrow \mathbf{X}_{\text{tr}}\mathbf{H}\mathbf{L}\mathbf{H}\mathbf{X}_{\text{tr}}^{\top}$
3: **Compute Dictionary:** $\mathbf{D} \leftarrow$ eigenvectors of $\boldsymbol{\Phi}$ corresponding to top k eigenvalues
4: **Compute Training Coefficients:** Replace $\mathbf{X}$ with $\mathbf{X}_{\text{tr}}$ in (2.3), use (2.3) to compute $\boldsymbol{\alpha}_{\text{tr}}$ given $\mathbf{D}$
5: **Compute Test Coefficients:** Replace $\mathbf{X}$ with $\mathbf{X}_{\text{ts}}$ in (2.3), use (2.3) to compute $\boldsymbol{\alpha}_{\text{ts}}$ given $\mathbf{D}$

there is exactly one nonzero element in each column $\mathbf{Y}$, where the position of the nonzero element indicates the class of the corresponding data sample. Similar to the previous case, the solution for the optimization problem given in (3.11) is the top eigenvectors of $\boldsymbol{\Phi} = \mathbf{X}\mathbf{H}\mathbf{L}\mathbf{H}\mathbf{X}^{\top}$. These eigenvectors compose the supervised dictionary to be learned. This dictionary spans the space where the dependency between data $\mathbf{X}$ and corresponding labels $\mathbf{Y}$ is maximized. The coefficients can be computed in this space using the *lasso* as given in (2.3). The optimization problem given in (3.11) compromises the reconstruction error to achieve a better discrimination power. In conclusion, the proposed supervised dictionary learning is given in Algorithm 1.

One important advantage of the proposed approach in Algorithm 1 is that the dictionary can be computed in closed form. Besides, learning the dictionary and the coefficients are performed separately, and it is not needed to learn these two iteratively and alternately, as is common in most supervised dictionary learning approaches in the literature (refer to Section 2.2).

3.3 Proposed Kernelized SDL

One of the main advantages of the proposed formulation for SDL, compared to other techniques in the literature, is that one can easily embed a kernel into the formulation. This enables nonlinear transformation of the data into a high-dimensional feature space where the discrimination of classes can be more efficiently performed. This is especially beneficial when incorporating data-dependent kernels[2], such as those based on normalized compression distance [34].

Kernelizing the proposed approach is straightforward. Suppose that $\mathbf{\Psi}$ is a feature map representing the data in feature spaces $\mathcal{H}$ as follows:

$$\begin{aligned} \mathbf{\Psi} : X &\rightarrow \mathcal{H} \\ \mathbf{X} &\mapsto \mathbf{\Psi}(\mathbf{X}). \end{aligned} \tag{3.12}$$

To kernelize the proposed SDL, it is sufficient to express the matrix of bases $\mathbf{U}$ as a linear combination of the projected data points into the feature space using representation theory [71], i.e., $\mathbf{U} = \mathbf{\Psi}(\mathbf{X})\mathbf{W}$. In other words, $\mathbf{W} \in \mathbb{R}^{n \times k}$ represents $\mathbf{U} \in \mathbb{R}^{p' \times k}$ in feature space $\mathbf{\Psi}(\mathbf{X}) \in \mathbb{R}^{p' \times n}$. Replacing $\mathbf{X}$ by $\mathbf{\Psi}(\mathbf{X})$ and $\mathbf{U}$ by $\mathbf{\Psi}(\mathbf{X})\mathbf{W}$ in the objective function of (3.11), one obtains

$$\begin{aligned} \mathrm{tr}(\mathbf{U}^\top \mathbf{\Psi}(\mathbf{X})\mathbf{HLH}\mathbf{\Psi}(\mathbf{X})^\top \mathbf{U}) &= \mathrm{tr}(\mathbf{W}^\top \mathbf{\Psi}(\mathbf{X})^\top \mathbf{\Psi}(\mathbf{X}) \\ &\quad \mathbf{HLH}\mathbf{\Psi}(\mathbf{X})^\top \mathbf{\Psi}(\mathbf{X})\mathbf{W}) \\ &= \mathrm{tr}(\mathbf{W}^\top \mathbf{KHLHKW}), \end{aligned}$$

with the constraint

$$\begin{aligned} \mathbf{U}^\top \mathbf{U} &= \mathbf{W}^\top \mathbf{\Psi}(\mathbf{X})^\top \mathbf{\Psi}(\mathbf{X})\mathbf{W} \\ &= \mathbf{W}^\top \mathbf{KW}, \end{aligned}$$

where $\mathbf{K} = \mathbf{\Psi}(\mathbf{X})^\top \mathbf{\Psi}(\mathbf{X})$ is a kernel function on data. Combining this objective function

[2] Although it is true that all kernels are computed on the data and hence, are data-dependent, the term is used in the literature to refer to those types of kernels that do not have any closed form.

and the constraint, the optimization problem for the kernelized SDL is

$$\begin{aligned} \max_{\mathbf{W}} \quad & \text{tr}(\mathbf{W}^\top \mathbf{KHLHKW}), \\ \text{s.t.} \quad & \mathbf{W}^\top \mathbf{KW} = \mathbf{I} \end{aligned} \tag{3.13}$$

whose solution is the top generalized eigenvectors of $(\mathbf{KHLHK}, \mathbf{K})$ according to the Rayleigh-Ritz Theorem [70]. To realize how the coefficients can be computed for the training and test sets, $\mathbf{U} = \mathbf{\Psi(X)W}$ can be replaced in (3.8), knowing that $\mathbf{X}$ has to be also replaced by $\mathbf{\Psi(X)}$, to obtain

$$\begin{aligned} \mathbf{V} &= \mathbf{W}^\top \mathbf{\Psi(X)}^\top \mathbf{\Psi(X)} \\ &= \mathbf{W}^\top \mathbf{K}. \end{aligned}$$

In other words, one can write

$$\mathbf{K} = (\mathbf{W}^\top)^\dagger \mathbf{V}, \tag{3.14}$$

where $(\mathbf{W}^\top)^\dagger$ is the left pseudoinverse of $\mathbf{W}^\top$. In (3.14), $\mathbf{V}$ includes the coefficients, and they can be made sparse by using the *lasso* as given in (2.3) with proper replacements, i.e., replacing $\mathbf{X}$ by the kernel over the training or test set, and $\mathbf{D}$ by the left pseudoinverse of $\mathbf{W}^\top$. Hence, the algorithm for kernelized SDL is given in Algorithm (2).

Algorithm 2 Kernelized Supervised Dictionary Learning

Input: Kernel on training data, $\mathbf{K}_{\text{tr}}$, kernel on test data, $\mathbf{K}_{\text{ts}}$, kernel on labels $\mathbf{L}$, training data size, n, size of dictionary, k.

Output: Dictionary, $\mathbf{D}$, coefficients for training and test data, $\boldsymbol{\alpha}_{\text{tr}}$ and $\boldsymbol{\alpha}_{\text{ts}}$.

1: $\mathbf{H} \leftarrow \mathbf{I} - n^{-1}\mathbf{e}\mathbf{e}^\top$

2: $\boldsymbol{\Phi} \leftarrow \mathbf{K}_{\text{tr}}\mathbf{H}\mathbf{L}\mathbf{H}\mathbf{K}_{\text{tr}}$

3: $\mathbf{W} \leftarrow$ top k generalized eigenvectors of $(\boldsymbol{\Phi}, \mathbf{K}_{\text{tr}})$

4: **Compute Dictionary:** $\mathbf{D} \leftarrow (\mathbf{W}^\top)^\dagger$

5: **Compute Training Coefficients:** Replace $\mathbf{X}$ with $\mathbf{K}_{\text{tr}}$ in (2.3), use (2.3) to compute $\boldsymbol{\alpha}_{\text{tr}}$ given $\mathbf{D}$

6: **Compute Test Coefficients:** Replace $\mathbf{X}$ with $\mathbf{K}_{\text{ts}}$ in (2.3), use (2.3) to compute $\boldsymbol{\alpha}_{\text{ts}}$ given $\mathbf{D}$

Chapter 4

Experiments on the Proposed SDL and KSDL

In this chapter, the performance of the proposed SDL is evaluated on various datasets and in different applications such as analyzing face data, digit recognition, and in classification of real-world data such as satellite images and biomedical data. The main advantages of the proposed SDL, such as a compact dictionary i.e., a discriminative dictionary even at small dictionary size and fast performance, will be shown through various experiments . In the next chapter, it will be shown how the kernelized version of the proposed SDL enables embedding data-dependent kernels into the approach to significantly improve the performance on difficult classification tasks. Table 4.1 provides the details of the datasets, their dimensionality, number of classes, and the number of instances per class, as well as in the training and test sets as being used in the experiments in this chapter.

4.1 Implementation Details

In the proposed approach, the first step is to compute the dictionary by computing the eigenvectors of $\mathbf{\Phi}$ as provided in Algorithms 1 or 2 for the proposed SDL or its kernelized version, respectively. To avoid rank deficiency in the computation of kernel on labels, the

Table 4.1: The datasets used in this chapter.

Dataset	Dataset Info.					
	Samples	Samples per Class	Training Size	Test Size	Classes	Dim.
Face (Olivetti)[a]	400	119, 281	200	200	2	4096
Digit (USPS)[b]	9298	-	7291	2007	10	256
Sonar[c]	208	97, 111	104	104	2	60
Ionosphere[c]	351	225, 126	176	175	2	34
Texture[d]	5500	500	2750	2750	11	40
Satimage[d]	6435	1533, 703, 1358, 626, 707, 1508	3218	3217	6	36
Heart[c]	297	160, 137	148	149	2	13
Parkinsons[c]	195	48, 147	97	98	2	22

[a]http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
[b]http://www-i6.informatik.rwth-aachen.de/~keysers/usps.html
[c]http://archive.ics.uci.edu/ml/
[d]http://www.dice.ucl.ac.be/neural-nets/Research/Projects/ELENA/databases/REAL/

identity matrix of the same size is added to the kernel, i.e., $\mathbf{L} = \mathbf{Y}^\top \mathbf{Y} + \mathbf{I}$. The next step is to calculate the coefficients in the *lasso* provided in (2.3). The GLMNET[1], which is an efficient implementation of the *lasso* using cyclic coordinate descent [39], is used. The optimal value of the regularization parameter in the *lasso* (λ^*), which controls the level of sparsity, has been computed by 10-fold cross-validation on the training set to minimize the mean-squared error. This λ^* is then used to compute the coefficients for both training and test sets[2].

For the first two datasets in Table 4.1, i.e., Olivetti Face and USPS Digit datasets, since intensity levels of pixels are used as features, the dimensionality of feature sets is high, i.e., 4096 and 256, respectively. In this high dimensional feature space, usually a linear kernel is sufficient to represent the data and hence, we have only provided the results for the proposed SDL with a linear kernel. The kernelized version of the proposed SDL do not usually help in this high dimensional feature space and hence, we have not provided the results for the proposed kernelized SDL (KSDL) for these two datasets. However, for other datasets in Table 4.1, we have also provided the results using an RBF kernel for the proposed KSDL.

As is suggested in [72], the coefficients computed on the training set are used for training a support vector machine (SVM). RBF kernel has been used for the SVM and the optimal parameters of the SVM, i.e., the optimal kernel width γ^* and the trade-off parameter C^*, are found by grid search and 5-fold cross-validation on the training set[3]. The coefficients computed on the test set are then submitted to this trained SVM to label unseen test examples.

Two measures are considered to evaluate the performance of the classification systems:

[1]The necessary tools and their Matlab interface can be accessed at http://www-stat.stanford.edu/∼tibs/glmnet-matlab/.

[2]GLMNET handles one data sample at a time and hence one λ^* is computed for each data point in the training set. However, the averaged λ^* over the whole training set is used to compute the coefficients on the training and test sets, as it yields better generalization.

[3]10-fold cross-validation yields very close results. Thus to avoid higher computation load, 5-fold cross-validation is adopted.

classification error and balanced classification error, which are defined as follows:

$$\mathrm{E} = \frac{n_{\mathrm{wr}}}{n}, \tag{4.1}$$

$$\mathrm{BE} = \frac{1}{c}\sum_{i=1}^{c}\frac{n_{\mathrm{wr}}^{i}}{n_i}, \tag{4.2}$$

where E and BE are classification error and balanced error, respectively; n_{wr} is the total wrongly classified data samples; n is the total number of data samples; c is the number of classes; n_{wr}^{i} is the number of wrongly-classified objects in class i; and n_i is the number of data samples in class i. According to this definition, E is the total number of wrongly-classified data samples over the total number of objects. Hence, if there are fewer objects in one class, wrongly-classified objects in that class contribute less towards the overall classification system error. The definition of BE, however, gives the same weight to all classes irrespective of the number of objects in each class. To further clarify the difference between these two measures, an extreme case example is provided. Suppose that in a two-class problem, there are 98 objects in one class and 2 objects in another class. If all 98 objects are correctly classified in class one, and out of two objects in class two, only one is correctly classified, the classification error is $\mathrm{E} = 1/100 = 1\%$, whereas the balanced error is $\mathrm{BE} = (1/2 + 0/98)/2 = 25\%$. If, for example, this classification system is supposed to classify healthy versus unhealthy cases, BE is a better measure to evaluate the classification system, because both classes equally contribute towards the estimation of error, irrespective of the number of data samples in each. Since as indicated on the third column of Table 4.1, some datasets used in the experiments, such as Face, Sonar, Ionosphere, and Satimage, are not balanced[4], both E and BE are provided for them in the next subsections.

[4]The USPS digit dataset is also somewhat imbalanced. However, since in the literature, particularly in [53] with which our results are compared, only classification error (E) is provided, our results are also presented using this measure only. Also since the publically available USPS data comes in separate training and test sets, and representing the number of instances per class takes space for 10 classes, this information is not provided for the USPS dataset in Table 4.1.

4.2 Face Data

In this experiment, the main goal is to show the compactness of the proposed dictionary. The Olivetti face dataset of AT&T [73] is used. This data consists of 400 face images of 40 distinct subjects, i.e., 10 images per subject, with varying lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). The original size of each image is 92×112 pixels, with 256 gray levels per pixel. However, in the experiments, each image has been cropped from the center to be 64×64 pixels.

The main task in these experiments is to classify the faces into glasses/no-glasses classes. To this end, the images are labeled to indicate these two classes, with 119 in the glass class and 281 in the no-glass. Typical images of these two classes are shown in Figure 4.1. All images are normalized to have zero mean and unit ℓ_2-norm. Half of the images are randomly selected for training, and the other half for testing; the experiments are repeated 10 times, and the average error (E) and balanced error (BE) are reported in Table 4.2. The experiments are performed on varying dictionary sizes, including 2, 4, 8, 16, and 32. The results are compared with several unsupervised and supervised dictionary learning approaches, as shown in Table 4.2. The dictionaries learned at the dictionary size of two are shown in Figure 4.2. For K-SVD, the fast implementation provided by Rubinstein [74] has been used. DK-SVD has been implemented using K-SVD as the core. The difference between supervised and unsupervised k-means is that in unsupervised k-means, the dictionary is learned on the whole training set, whereas in the supervised one, one dictionary is learned per class as suggested in the texton-based approach by Varma and Zisserman [19, 44]. The code for metaface approach has been provided by the authors [45]. The same as the proposed approach, the parameter(s) of all these rival approaches are tuned using 5-fold cross-validation on the training set.

As can be seen in Table 4.2, our approach performs the best among these approaches. The compactness of the dictionary learned using the proposed SDL is noticeable from the results at small dictionary size. For example, at the dictionary size of two, while the error of our approach is 12.8%, unsupervised k-means yields a 27.4% error, which is more than twice as large as our approach. The best result obtained by other supervised dictionary approaches (here metaface) yields a 17.55% error at this dictionary size, which is about

Figure 4.1: Typical face images from the Olivetti face dataset in two classes of glasses vs. no-glasses.

5% above the error generated by the proposed SDL. The same conclusion can be made using balanced error. Interestingly, supervised k-means performs significantly better than the unsupervised one, particularly at small dictionary sizes. Also as can be seen from Figure 4.2, the dictionaries learned using supervised approaches (Figures 4.2c to 4.2f) are more discriminative on the classification task, which is distinguishing between the faces with and without glasses, than the dictionaries learned using unsupervised approaches (Figure 4.2a and 4.2b). The main conclusion of this experiment is that the proposed SDL generates a very discriminative and compact dictionary, compared to well-known unsupervised and supervised dictionary learning approaches.

4.3 Digit Recognition

The second experiment is performed on the task of handwritten digit classification on the USPS dataset [75]. As most of the supervised dictionary learning approaches show results on this dataset, the proposed approach is applied on the same data for the purpose of comparison with existing approaches. This dataset consists of handwritten digits, each with the size of 16×16 pixels with 256 gray levels. There are 7291 and 2009 digits in the training and test sets, respectively.

Table 4.2: Classification error (E) and balanced error (BE) on test set for Olivetti face data using the proposed SDL. The results are compared with several other dictionary learning approaches in the literature. The best results obtained are highlighted.

Approach		Dictionary Size									
		2		4		8		16		32	
		E	BE	E	BE	E	BE	E	BE	E	BE
Unsupervised	k-means	27.40	39.35	22.60	29.05	13.15	17.36	8.15	10.71	5.75	8.40
		±2.04	±4.01	±5.18	±5.97	±2.38	±4.17	±1.81	±3.12	±1.70	±2.62
	K-SVD [37]	28.20	41.40	20.60	27.36	9.65	12.97	7.75	11.07	4.05	6.06
		±2.45	±4.60	±2.41	±5.23	±1.62	±2.52	±2.06	±3.13	±1.23	±2.26
Supervised	Proposed SDL	**12.80**	**16.61**	**10.05**	12.46	**4.95**	**5.77**	**4.95**	**5.91**	**3.30**	**4.41**
		±3.77	**±4.68**	**±3.11**	±4.40	**±1.92**	**±2.94**	**±1.14**	**±1.98**	**±1.53**	**±2.28**
	DK-SVD [50]	17.80	19.36	10.25	**10.15**	8.75	11.25	7.05	8.70	6.75	10.13
		±3.06	±3.52	±2.48	**±3.04**	±2.02	±4.06	±2.11	±1.31	±1.53	±2.92
	k-means[a] [19]	17.75	23.45	10.40	14.01	7.40	10.57	5.55	7.84	3.65	5.45
		±3.65	±5.71	±2.56	±2.76	±1.90	±2.93	±1.62	±2.58	±1.20	±2.01
	Metaface [45]	17.55	19.39	11.25	15.35	9.75	14.58	7.60	11.74	5.45	9.28
		±2.87	±3.02	±2.35	±2.61	±3.58	±5.88	±1.39	±1.91	±0.96	±1.46

[a]Supervised k-means learns one sub-dictionary per class and then compose all learned sub-dictionaries into one.

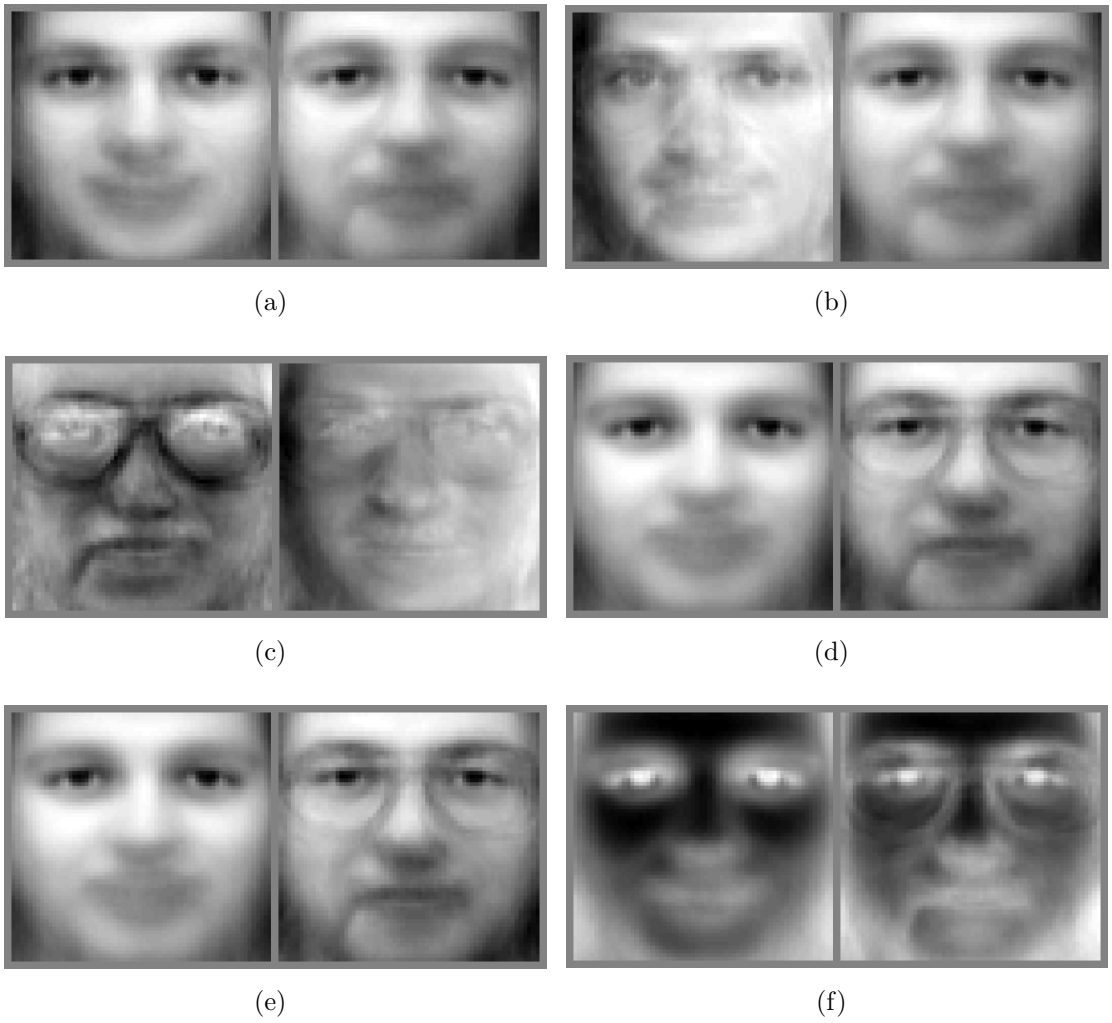


Figure 4.2: The dictionaries learned at the dictionary size of two on the Olivetti Face dataset for (a) unsupervised k-means, (b) unsupervised K-SVD, (c) proposed SDL, (d) DK-SVD, (e) supervised k-means, and (f) metaface.

Our results are compared with the most recent SDL technique, which yields the best results published so far on this dataset [53]. To facilitate a direct comparison with what is published in [53], the same setup as they have reported is used. To this end, since the most effective techniques on digit recognition deploy shift invariant features [76], and

since neither our approach nor the one reported in [53] benefit from these kind of features, as suggested in [53], the training set is artificially augmented by adding digits which are shifted version of original ones, moved by one pixel in all four directions. Although this is not an optimal and sophisticated way of introducing shift invariance to the SDL techniques, it takes into account this property in a fairly simple approach. Each digit in training and test sets is normalized to have zero mean and unit ℓ_2-norm.

Table 4.3 shows the results obtained using the proposed approach in comparison with the unsupervised and supervised dictionary learning techniques reported in [53]. As can be seen, again our approach introduces a very compact dictionary such that its performance at dictionary size of 50 is the same as the performance of the system reported in [53] using a dictionary of 100 atoms. With increasing the dictionary size, the performance of our approach slightly degrades. This is mainly because the bases or dictionary atoms in our approach are associated with the directions of maximum separability of the data, as has been enforced by the optimization problem in (3.11). Nevertheless, the number of useful bases depends on the intrinsic dimensionality of the subspace, which in turn depends on the nature of the data. If the number of dictionary atoms goes beyond this intrinsic dimensionality, then adding more atoms does not improve the performance but may degrade it, as they are not associated with the separable directions but related to noise. On the other hand, it is important to notice that a reasonable performance using much less complexity than the best rival can be achieved. It should be also noted that the best performance achieved by our approach (happening at a small dictionary size of 50) is just 0.25% worse than the best results obtained by [53] (happening at dictionary size of 300, i.e., with much higher complexity). This means that our approach misclassifies only 5 more digits compared to the best results obtained in [53], whereas for the same dictionary size (50), our approach performs 0.55% better, i.e., classifies 11 more digits correctly. On the other hand, with respect to the complexity, our proposed approach offers a much simpler solution for SDL than the approach in [53]: there are fewer parameters to tune, the dictionary can be computed in closed form, and there is no need to solve a complicated nonconvex optimization problem as is used in [53] by iteratively and alternatively optimizing classifier, dictionary, and coefficients learning.

As a final remark, due to the orthonormality constraint in the optimization problem

Table 4.3: Classification error on test set for digit recognition on USPS data using proposed SDL compared with the most effective SDL approach reported in the literature on the same data [53]. Highlighted entries represent the best results obtained at each dictionary size.

Approach	Dictionary Size			
	50	100	200	300
Proposed SDL	**3.09**	3.19	3.64	-
Unsupervised [53]	8.02	6.03	5.13	4.58
Supervised [53]	3.64	**3.09**	**2.88**	**2.84**

of our proposed SDL as given in (3.11), overcompleteness is not possible in our proposed SDL. This is the reason that in Table 4.3, no results are reported for a dictionary size of 300 for our approach. However, as mentioned above, due to the compactness of our dictionary, good results are obtained at a much smaller dictionary size, which is a desired attribute as it decreases the computational load. Also, the proposed kernelized version of our proposed approach given in (3.13) and Algorithm 2 can learn dictionaries as large as n, i.e., the number of data points used for training, which is usually greater than the dimensionality of the data p (see Table 4.1 for the relative size of p and n for the data used in our experiments).

4.4 Other Real-World Data

In the two previous sections, the classification task was performed on the pixels of images directly. In this section, the performance of the proposed approach is evaluated on the classification of some real-world data represented using features extracted. Six datasets with varying complexity from 2- to 11-class, with the dimensionality of up to 60 features, and also with as many as 6435 data samples are used in these experiments (refer to Table 4.1 for detailed information on these datasets). Sonar, Ionosphere, and Texture datasets are preprocessed to have zero mean and unit ℓ_2-norm, whereas for the Heart, Parkinsons, and Satimage datasets, the features are normalized to be in the range of [0, 1] due to the large variation of their feature values.

Since the rival approaches are the same as that used for face data, their implementations are the same as was explained in Section 4.2. There is one additional remark here, on the implementation of supervised k-means on datasets with more than two classes, such as the Texture and Satimage datasets. This approach has been implemented in a way to ensure that the dictionary atoms are evenly distributed over different classes as much as possible. For example, in the case of dictionary size of 8 and for the Texture dataset that has 11 classes, first 11 dictionary atoms are selected, one from each class, then 8 of them are randomly retained and the rest are removed.

On all datasets, the experiments are repeated ten times over a random split of data into half for training and half for testing. The average and standard deviation of classification error (E) and balanced error (BE) are reported in Tables 4.4 to 4.7, in comparison with several other unsupervised and supervised dictionary learning approaches. Since the texture dataset is balanced, the error and balanced error are the same, therefore, the balanced error has not been reported for this dataset in Table 4.5. The results of classification are also included using a kernelized version of the proposed SDL with radial basis function (RBF) as the kernel. The width of the RBF kernel has been selected based on a self-tuning approach [77].

As can be seen from Tables 4.4 to 4.7, the proposed SDL or its kernelized version performs the best in all cases, except on the Sonar dataset at the dictionary sizes of 8 and 16. The better performance of supervised k-means at the dictionary sizes of 8 and 16 on the Sonar data is not significant, as the resultant standard deviation is very high. DK-SVD performs poorly (even worse than the unsupervised K-SVD approach) on all these datasets mainly because, by design, it uses a linear classifier (refer to Section 2.2 and [50] for more detail on this approach). The poor performance of metaface is because it usually performs well at very large dictionary size. Hence, at reported dictionary sizes, its training is not sufficient to capture the underlying data structure. For example, for Sonar data, while the proposed SDL can achieve an error of 20.77 ± 4.67 at the dictionary size of 32, the metaface approach can only achieve this accuracy at the dictionary size of 64 (error 20.00 ± 4.75). However, using large dictionary size adds to the computational load of the approach.

As a final remark on the results presented in this subsection, I would like to comment on the relative performance of the proposed SDL and its kernelized version KSDL. The

relative performance of these two approaches mainly depends on the nature of the data to be classified, and whether it has a linear or nonlinear behavior. In other words, it depends whether the data can be represented as a subspace or a submanifold. In the former case, the proposed SDL should be sufficient to model the data, while in the latter case, the KSDL should potentially perform better. However, the success of KSDL depends on the proper selection of the kernel and its parameter(s). In fact, even if the data has a linear nature and can be represented in a subspace, the KSDL should also perform as well as SDL, but this again depends on proper kernel and model selection.

4.5 Computation Cost of the Proposed SDL

The proposed SDL is computationally fast and efficient due to three main attributes of the approach, i.e., a compact dictionary, a closed form dictionary, and few parameters to tune.

As related to the first attribute, i.e., compact dictionary, it was shown using many experiments that the proposed approach almost always (except on the Sonar data for which the better performance of supervised k-means is not significant) performs better than other unsupervised and supervised dictionary learning approaches. For example, on digit recognition, it was shown that the approach at the dictionary size of 50 performs the same as the SDL proposed in [53] using a dictionary size of 100. Performing at the same level as another SDL approach using half the dictionary size, significantly saves the computational cost. This can also be observed on face and some other real-world datasets.

With regards to the second attribute, as elaborated in Section 2.2, most unsupervised and supervised methods are based on the *lasso* formulation, and since the formula is nonconvex when both dictionary and coefficients are unknown, it has to be solved iteratively and alternately. For the third category of SDL, for which the classifier has to also be learned in the optimization problem, learning the classifier is also added to this list. In addition to the possibility of getting stuck in local minima, this makes the computation expensive. The computation load will be even higher considering that there are several parameters in proposed SDL approaches in the literature that must be tuned by cross-validation on the training set. Our approach, proposes a closed form for the dictionary. Except for

Table 4.4: The results of classification *error* (%) on Sonar, Ionosphere, and Texture datasets using different methods and in different dictionary sizes. The best results obtained are highlighted.

Approach		Sonar			Ionosphere			Texture		
		8	16	32	8	16	32	8	16	32
Unsupervised	k-means	28.56	24.52	24.42	7.37	7.71	8.06	2.49	1.12	0.97
		±5.53	±5.43	±3.77	±2.48	±1.41	±1.86	±0.66	±0.25	±0.29
	K-SVD [37]	27.31	24.81	28.56	8.69	9.09	8.00	1.54	0.81	0.83
		±2.69	±6.69	±4.25	±4.12	±1.73	±1.50	±0.30	±0.27	±0.19
Supervised	Proposed SDL	27.79	22.50	**20.77**	5.94	**5.60**	**5.43**	**1.44**	**0.45**	**0.31**
		±3.47	±2.73	**±4.67**	±1.66	**±1.41**	**±1.41**	**±0.38**	**±0.12**	**±0.10**
	KSDL-RBF[a]	28.75	27.31	26.35	**5.66**	5.89	6.17	1.68	1.20	1.19
		±3.88	±4.40	±3.22	**±1.97**	±2.03	±2.07	±0.26	±0.23	±0.22
	DK-SVD [50]	32.40	32.69	29.04	16.11	18.00	15.89	27.91	6.15	7.28
		±4.53	±4.32	±4.15	±1.88	±3.51	±2.50	±3.87	±0.82	±1.86
	k-means [19]	**24.62**	**22.31**	22.88	7.54	9.54	10.00	2.11	0.95	0.82
		±5.31	**±4.27**	±5.98	±1.39	±1.59	±2.35	±0.42	±0.14	±0.22
	Metaface [45]	26.74	27.89	23.17	18.29	21.54	16.29	9.76	10.03	4.64
		±3.17	±5.22	±4.43	±1.62	±2.89	±2.52	±0.55	±1.88	±0.57

[a]Proposed kernel SDL with RBF kernel.

Table 4.5: The results of classification *balanced error* (%) on Sonar and Ionosphere datasets (Texture data is not included because the error and balanced error are the same as the dataset is balanced) using different methods and in different dictionary sizes. The best results obtained are highlighted.

Approach		Sonar			Ionosphere		
		8	16	32	8	16	32
Unsupervised	k-means	28.41	24.26	24.25	8.36	9.40	9.25
		±5.66	±4.93	±3.97	±3.11	±1.56	±2.97
	K-SVD [37]	27.09	24.87	28.73	10.35	10.63	8.46
		±2.61	±6.36	±4.27	±4.97	±2.05	±1.60
Supervised	Proposed SDL	27.75	22.65	**20.86**	7.06	**6.22**	**6.20**
		±3.50	±2.87	**±4.84**	±1.57	**±1.39**	**±1.62**
	KSDL-RBF[a]	28.64	27.12	26.03	**6.06**	6.33	6.58
		±3.91	±4.79	±3.28	**±1.92**	±2.13	±2.20
	DK-SVD [50]	33.61	33.79	29.56	18.98	20.17	18.78
		±3.76	±4.46	±4.11	±3.54	±4.67	±4.49
	k-means [19]	**24.74**	**22.32**	22.48	8.61	11.29	11.79
		±5.06	**±4.26**	±5.65	±1.92	±1.94	±3.34
	Metaface [45]	27.35	30.20	23.76	25.79	29.67	24.43
		±4.13	±3.44	±6.19	±2.98	±3.64	±3.32

[a]Proposed kernel SDL with RBF kernel.

Table 4.6: The results of classification *error* (%) on Heart, Parkinsons, and Satimage datasets using different methods and in different dictionary sizes. The best results obtained are highlighted.

Approach		Heart			Parkinsons			Satimage		
		4	8	12	4	8	16	8	16	32
Unsupervised	k-means	23.92	22.23	21.42	14.95	12.89	13.81	13.36	13.02	12.87
		±1.86	±2.37	±2.68	±3.80	±4.33	±3.51	±0.47	±0.64	±0.72
	K-SVD [37]	19.80	21.42	18.99	12.27	12.89	13.30	10.42	10.70	11.92
		±2.67	±2.57	±2.57	±3.99	±4.43	±3.72	±0.43	±0.73	±0.36
Supervised	Proposed SDL	26.49	25.07	25.20	13.30	9.90	10.10	11.25	10.58	10.66
		±3.31	±5.18	±5.57	±3.75	±4.77	±3.97	±0.36	±0.40	±0.41
	KSDL-RBF[a]	**18.51**	**16.55**	**16.89**	**12.06**	**9.18**	**9.48**	**10.18**	**9.81**	**9.66**
		±3.07	**±2.45**	**±1.88**	**±4.18**	**±3.59**	**±3.46**	**±0.36**	**±0.38**	**±0.33**
	DK-SVD [50]	21.55	23.38	26.62	20.52	17.94	21.44	35.36	20.15	28.89
		±2.12	±3.88	±3.72	±5.33	±7.61	±3.85	±13.29	±1.38	±4.19
	k-means [19]	21.42	20.88	19.53	17.37	13.92	10.62	13.61	12.65	12.98
		±3.59	±2.31	±3.41	±3.69	±3.64	±3.00	±0.36	±0.32	±0.65
	Metaface [45]	20.00	23.99	24.66	18.56	23.20	16.39	23.43	27.14	24.85
		±2.59	±3.52	±3.90	±4.01	±4.84	±3.72	±1.38	±1.05	±1.53

[a]Proposed kernel SDL with RBF kernel.

Table 4.7: The results of classification *balanced error* (%) on Heart, Parkinsons, and Satimage datasets using different methods and in different dictionary sizes. The best results obtained are highlighted.

Approach		Heart			Parkinsons			Satimage		
		4	8	12	4	8	16	8	16	32
Unsupervised	k-means	23.95	22.49	21.48	22.77	18.42	21.36	16.99	16.39	16.50
		±2.04	±2.39	±2.74	±7.27	±8.51	±8.27	±0.65	±0.86	±0.73
	K-SVD [37]	19.94	21.83	19.13	18.69	21.34	22.46	13.31	13.44	15.16
		±2.52	±2.60	±2.47	±7.17	±8.25	±5.96	±0.54	±0.85	±0.48
Supervised	Proposed SDL	26.72	25.30	25.56	21.88	15.34	16.16	14.07	13.12	13.25
		±3.58	±5.08	±5.37	±6.30	±7.70	±8.31	±0.33	±0.54	±0.49
	KSDL-RBF[a]	**18.72**	**16.74**	**17.07**	**19.08**	**14.78**	15.21	**12.96**	**12.32**	**12.14**
		±3.21	**±2.46**	**±1.98**	**±8.34**	**±6.46**	±7.12	**±0.40**	**±0.60**	**±0.46**
	DK-SVD [50]	22.15	23.30	26.40	27.74	27.85	26.98	35.64	22.22	29.85
		±2.05	±3.69	±3.96	±4.52	±7.85	±4.28	±8.80	±1.51	±3.42
	k-means [19]	21.33	20.99	19.66	26.29	21.80	**14.75**	17.21	16.01	16.55
		±3.38	±2.33	±3.26	±6.42	±6.60	**±5.44**	±0.48	±0.52	±0.88
	Metaface [45]	20.06	24.01	24.62	24.62	28.13	25.24	30.26	32.89	32.26
		±2.63	±3.44	±3.59	±4.37	±3.37	±5.19	±1.94	±2.01	±2.27

[a]Proposed kernel SDL with RBF kernel.

unsupervised and supervised k-means, none of the other dictionary learning approaches benefit from a closed-form dictionary. Note that even in k-means, the cluster centers have to be found using an iterative approach, and the dictionary does not have a closed form. However, the approach does not have to iteratively and alternatively learns the dictionary and coefficients, i.e., the dictionary can be computed first and then the coefficients are learned. Hence, except k-means, the other approaches are slower than the proposed SDL from this point of view.

As was explained in the second attribute of the proposed SDL above, usually dictionary learning approaches have several parameters to be tuned. Because these parameters are to be tuned within an iterative method, their fine tuning is very time-consuming. This is especially elaborated in [16, 53] (refer to the part related to parameter tuning in the "Experimental Validation Section" of these two papers). In our approach, the only parameter to tune is λ, and this can be quickly tuned as the dictionary is computed in closed form.

Table 4.8 provides the computation time (in seconds) using an 8-core machine on the Sonar and Heart datasets for one iteration. Only one core is used in these computations and no parallel processing is performed. The computation time includes all processing necessary for classification, such as tuning the parameters, training, and testing. As can be observed from Table 4.8, the computation time for the proposed approach is not significantly increased with increasing the dictionary size. This can be related to the number of directions of separation, which is data dependent. If sufficient number of these directions are already included, with increasing the number of atoms in the dictionary, the number of nonzero coefficients are not very much increased and hence, the computation time for the training and testing the classifier remains almost the same. In other words, the additional computation time is just due to the computation of more dictionary atoms and corresponding coefficients.

The relative magnitude of computation time for other datasets is almost the same, i.e., our proposed approach is the fastest, then k-means, followed by DK-SVD and K-SVD. Metaface is the slowest partly due to its non-optimized implementation, and partly due to its intrinsic high computational load.

Table 4.8: The average computation time per fold (in seconds) on Sonar and Heart datasets for various unsupervised and supervised dictionary learning approaches. The computation is performed using an 8-core machine (only one core has been used in the computation and no parallel processing is used).

Approach		Sonar			Heart		
		8	16	32	4	8	12
Unsupervised	k-means	3.23	7.67	29.00	5.14	6.14	15.94
	K-SVD [37]	22.24	36.88	55.72	51.33	35.12	58.87
Supervised	Proposed SDL	2.06	2.39	3.83	4.41	4.51	6.58
	KSDL-RBF[a]	2.12	2.44	3.33	0.35	0.34	0.36
	DK-SVD [50]	13.42	32.78	58.52	11.61	20.96	44.60
	k-means [19]	3.12	8.48	31.03	4.25	6.08	21.71
	Metaface [45]	496.5	883.89	1937.87	444.92	969.41	1638.07

[a]Proposed kernel SDL with RBF kernel.

4.6 The Effect of Noisy Labels on the Performance of the Proposed SDL

Since in supervised dictionary learning approaches the information category is used in the learning of the dictionary, one main question will be: "to what extent are these approaches sensitive to noisy labels?". In this subsection, it is tried to address this question.

As defined in Section 3.2, the labels $\mathbf{Y} \in \{0,1\}^{c\times n}$ can only take the values 0 or 1. Therefore, what is meant by noisy labels is that 0 might be converted to 1, or vice versa. It is assumed that in each column of noisy labels $\hat{\mathbf{Y}}$, there is still only one nonzero element, which indicates the class of the corresponding object.

Almost all the categories of SDL mentioned in Section 2.2 utilize the labels directly or indirectly in the learning of the dictionary. For example, in the first SDL category, one dictionary is learned per class. Therefore, if one object is wrongly assigned to a class, this object will contribute to learning dictionary atoms in the wrong class, which consequently

may lead to reducing the efficiency of the learned dictionary in the classification task. In our proposed approach, as indicated in the optimization problem (3.11), a linear kernel over the labels is used to include the category information in the learning of the dictionary. Hence, it is natural that one expects that noisy labels degrade the efficacy of the dictionary learned in the classification task.

To address the question raised in the beginning of this subsection, some experiments are performed on the Olivetti face dataset. In these experiments, a certain percentage of wrong labels are included into the learning of the supervised dictionary, and then the classification task is performed using this dictionary. Since the main concern is to see how sensitive the dictionary is to noisy labels, correct labels are used in the classifier over the training set. In other words, in our experiments, noisy labels are only used in the learning of the dictionary, and correct labels in the classifier. This may not be a realistic setup as when there are wrong labels, the assumption is that correct labels are not available, otherwise one could also use them in the learning of the dictionary. However, if wrong labels are used in the classifier as well, one do not know to what extent the dictionary is affected by wrong labels because wrong classification might be also due to misguiding the classifier.

The results are shown in Figure 4.3 for the dictionary sizes of 2, 4, 8, and 16, and for various supervised dictionary learning approaches as used in the experiments on the Olivetti face dataset (refer to Table 4.2). As can be seen from this figure, our proposed SDL is the least sensitive one to noisy labels. Also, by increasing the dictionary size, the sensitivity to noisy labels is reduced for our proposed SDL as well as for the supervised k-means. It makes sense to see lower sensitivity to noisy labels at larger dictionary sizes for the proposed SDL, because noisy labels will cause the discriminative directions to move away from leading atoms or bases in the learned dictionary, which degrades the effectiveness of the dictionary at small dictionary sizes, while at larger dictionary sizes, these discriminative directions will appear again, although not in leading atoms. Also in supervised k-means, by increasing the dictionary size, it is more likely that some of the cluster centers in each class, which are the dictionary atoms in that class, correspond to the correctly-labeled data samples. For example, if the dictionary size is two in a two-class problem, there is only one dictionary atom per class. Hence, if this dictionary atom represents wrong data samples

due to noisy labels, the dictionary learned completely fails to model the data samples correctly. However, by increasing the dictionary size, this catastrophic failure is less likely to happen.

However, this phenomenon cannot be observed for the DK-SVD and the metaface approaches. DK-SVD does not follow this behavior mainly because the learning of the dictionary and classifier is performed in one optimization problem, as explained in Subsection 2.2.3 and in [50]. Hence, noisy labels also affect the learning of the linear classifier involved, and I could not find any way to include the noisy labels only in the learning of the dictionary, not the classifier.

Similarly, in the metaface approach, the class labels used during learning the dictionary are used to tag each dictionary as to what class it belongs to. This tag is later used to indicate the class label of the test object that minimizes the residue obtained using the reconstruction error computed on the subdictionary elements belonging to a class and a test object. Therefore, similar to the DK-SVD approach, there is no way to include the noisy class labels only in the learning of the dictionary, and not in the classifier. This explains why noisy labels have greater impact on DK-SVD and metaface approaches, as they affect both the dictionary learning and training of the classifiers. Based on these explanations, I admit that comparing the effect of noisy labels on the proposed SDL with DK-SVD or metaface is not completely fair, as in the proposed approach (as well as in supervised k-means) the impact of noise on the training of classifiers was deliberately avoided, whereas there was no way to avoid it in the DK-SVD and metaface approaches.

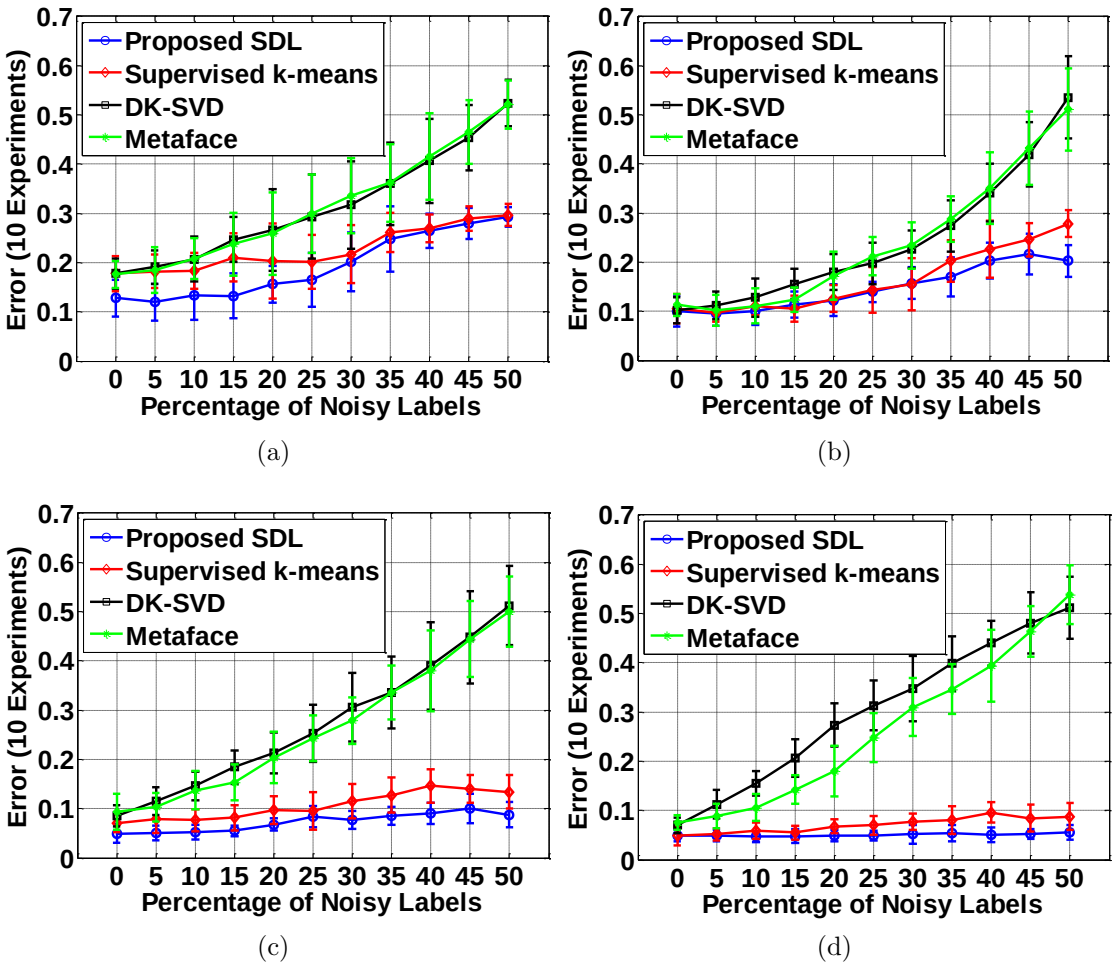


Figure 4.3: The error rate of the classification system for Olivetti face recognition system to discriminate between faces with and without glasses. The effect of noisy labels in learning dictionary are shown for the dictionary sizes of (a) 2, (b) 4, (c) 8, and (d) 16.

Chapter 5

Data-Dependent Kernels

As was shown in Section 3.3, one of the main attributes of the proposed supervised dictionary learning (SDL) approach is that it can be easily kernelized. This makes it possible that the proposed SDL can benefit from data-dependent kernels, which are specifically designed for the application in hand.

In this chapter, a novel compression-based (dis)similarity measure is proposed. This measure is computed using MPEG-1, a lossy compressor for 2D data.The properties of the proposed measure will be investigated, i.e., it will be shown that it properly works on both small and large patch sizes of texture images. Eventually, it will be demonstrated that embedding the proposed measure into the kernelized SDL introduced earlier in Section 3.3 significantly improves the performance of classification systems in texture analysis applications.

5.1 Introduction

Objects can be represented by either features or (dis)similarities. In feature-based representation, a set of measurements (features) are computed based on expert knowledge of the domain. In (dis)similarity-based representation, objects are represented by their pairwise comparisons [78]. (Dis)similarity-based representation can be computed either on features,

or by comparing objects directly using a (dis)similarity measure [79]. There are cases, however, that computing descriptive features to represent objects for a specific learning task is difficult or impossible due to insufficient knowledge of the domain [80]. For instance, as texture is a complicated phenomenon, there is no definition that is agreed upon by the researchers in the field [81]. This is one of the reasons that there are various feature-based techniques in the literature, each of which tries to model one or several properties of textures depending on the application in hand. The performance of each of these features depends on the texture type, and there is no single feature method that performs well on all different texture types [82, 83]. To avoid this problem, objects, for example textures in this case, can be represented in (dis)similarity space. In this approach, pairs of texture patches are compared by a (dis)similarity measure reflecting their mutual resemblance.

Early twenty first century, a similarity measure based on Kolmogorov complexity, i.e., so-called normalized information distance (NID), has been introduced in the literature with attractive attributes such as being metric, parameter-free, and universal [84]. It is parameter-free in the sense that it does not use any feature or background knowledge about the data, and universal as it minorizes all other distances. However, due to non-computability of Kolmogorov complexity, it has been mainly approximated using either real-world lossless compressors [34] or by the entropy rate [85]. The former introduced normalized compression distance (NCD), and the latter led to the introduction of normalized entropy-rate distance (NED). Intuitively, NCD is a compression-based similarity measure, i.e., two objects are considered close or similar if one can be significantly compressed by providing the information in the other. NCD has been successfully utilized in many applications such as alignment-free whole genome phylogeny, chain letter history, language history, plagiarism detection, music classification and clustering, protein sequence classification, and heart rhythm analysis (refer to [86, 87] for a list of applications).

5.2 Normalized Information Distance

Normalized information distance (NID) [84] is a metric and a universal parameter-free similarity measure based on Kolmogorov complexity that minorizes all other distance mea-

sures [84]. Intuitively, NID is a measure of the complexity of objects by finding their shortest description. If the shortest description of an object is long, the object is considered a complex object.

To understand the definition of the NID, one needs to define two notations: $K(x)$ and $K(x|y)$. The former is the Kolmogorov complexity of string x, which is defined as the length of the shortest binary program to compute x on a universal computer such as a universal Turing machine, whereas the latter is the conditional Kolmogorov complexity, which is defined as the length of the shortest program to compute x if y is provided as an auxiliary input for the reference [84]. The NID is defined as

$$NID(x,y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}}. \tag{5.1}$$

5.3 Normalized Compression Distance

Since Kolmogorov complexity is a noncomputable measure, the NID defined in (5.1) is computed by approximating Kolmogorov complexity using a compressor denoted by C as follows [34]

$$NCD(x,y) = \frac{\min\{C(xy), C(yx)\} - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}, \tag{5.2}$$

where xy means that the strings x and y are concatenated. While NID is a universal measure, i.e. it minorizes all other distances, NCD inherits this property from NID to some extent - but not fully. In other words, NCD is quasi-universal; since NID is not computable, it is impossible to estimate how far NCD is from its ideal counterpart NID [34].

To have more insight into (5.2), the case is considered that $C(y) \geq C(x)$[1] and the compressor is symmetric such that $C(xy) = C(yx)$. In this case, (5.2) can be rewritten as $NCD(x,y) = \frac{C(xy)-C(x)}{C(y)}$, which means that the NCD distance between x and y is an improvement on compressing y using x (the numerator, which is also denoted as $C(y|x)$) over compressing y by its own (the denominator) [34]. This interpretation will help to explain our proposed measure later in next subsection.

[1]The opposite condition can be interpreted similarly, as the NCD distance defined in (5.2) is symmetric.

5.4 1D vs. 2D Compressors

NCD was initially defined on binary strings and one-dimensional (1D) data with the explanation that all data can be converted to this kind of representation. Also, lossless compressors were originally used in NCD, as they satisfy the conditions required for a compressor to be a normal compressor as defined in [34], using which it can be proved that NCD is a metric. Further, a lossless compressor can adequately explore repeated structures in 1D data, which are needed for finding similarities among strings [88]. Many of the initial applications of NCD were based on 1D data, as can be seen from typical NCD applications listed earlier in Section 5.1.

However, there has been a high demand for using (dis)similarity-based representation in 2D data such as images,[2] and the extension of NCD to 2D data does not seem to be straightforward for several reasons. First, converting 2D data to 1D does not fully preserve the spatial locality and connectivity of neighboring pixels. Second, there is no single approach to convert 2D to 1D data as there are two degrees of freedom in this conversion: how to scan an image and how to take the pixels from two images whose similarities are to be computed [85]. The first degree of freedom leads to many different approaches of linearizing 2D data such as row by row, column by column, scan filling curves, global line sampling, zigzag linearization, Hilbert-Peano curve, and self-describing context-based pixel ordering (SCPO) [85, 89]. The effect of linearization using four of the above linearization approaches was empirically investigated on the computation of NCD in [89]. They concluded that: "images may not be fully expressible as a string, at least using current compression algorithms" and also that their results "certainly indicate that the method of linearization does matter". This is confirmed by [90], as row by row linearization produces better results than column by column in color image retrieval on the Corel dataset, which means that the method of linearization has an impact on the results, which is not desirable. As for the second degree of freedom, it is possible that the pixels from two images are taken alternately, or all pixels are scanned from the first image and then from the other. In another attempt to encode the spatial locality in converted

[2] In this chapter, our application is mainly on images and therefore, from now on, whenever 2D data is mentioned, images are meant.

2D data to strings, it was proposed in [91] to add an extra bit when converting images to strings to consider the vertical connection of pixels, but the improvement in the results was almost negligible.

The next step was to use JPEG and JPEG 2000 as 2D compressors to compute NCD-based measures. However, this led to contradictory results in the literature: while it is shown in [92] that using JPEG2000 yields better results than string compressors on the classification of satellite images, [89] and [93] conclude that JPEG and JPEG2000 does not yield better results than string compressors on images. Also, Bardera *et al.* conclude that string compressors perform better than an image compressor (using JPEG2000) in image registration [85].

An alternative approach is using MPEG encoders as 2D compressors in NCD. The main advantage of an MPEG compared to a JPEG encoder is that while JPEG is designed for compressing one image, MPEG encodes frames of images and hence, by considering two images as two frames, they can be compressed in reference to each other, a desirable trait in NCD. In the next section, a novel formulation based on the MPEG encoder is proposed for measuring (dis)similarity between images/patches. It will be shown that this new measure works well on both small and large patch sizes. Introducing this new measure in this thesis, it will be also shown that the results of pixel-based texture classification can be significantly improved compared to other NCD-based approaches in the literature.

5.5 Proposed Compression-Based Similarity Measure

Since an MPEG-1 encoder is used in the proposed (dis)similarity measure, some description of how this encoder works will be first provided.

MPEG-1 is a 2D encoder and thus, it takes into account the spatial locality and connectivity of the neighboring pixels in images for compression. MPEG-1 was originally designed for compressing movies based on three different coding schemes, i.e., intra-frame (I-frame) coding, predictive frame (P-frame) coding (also called inter-frame coding), and bidirectional frame (B-frame) coding [94]. I-frame coding is performed on individual frames without reference to other frames using discrete cosine transform (DCT). P-frame codes

a frame in reference to the previous one by using a block-matching algorithm for motion estimation and using DCT on the residual. Finally, B-frame coding compresses a frame with reference to its next and previous frames. To utilize MPEG-1 as the compressor in compression-based similarity measures, patches/images are considered as two successive frames and compressed using the MPEG-1 encoder. This avoids the need to linearize the images, which causes loss of spatial locality, as explained in the previous section. Since there are only two frames (two images whose similarity are to be computed), B-frame coding is not utilized.

To use MPEG-1 as the compressor in a (dis)similarity measure, one needs to use a proper formulation based on how MPEG-1 works. To this end, based on the description provided above for the MPEG-1 encoder, and also the explanation of (5.2) at the end of Section 5.3, the new dissimilarity measure is proposed considering these two points: first, MPEG-1 is utilized for the computation of $C(x|y)$ (the conditional compression of x given y) using only P-frame coding - and I-frame coding is bypassed as it does not provide any information on the similarity of x and y - and it is denoted as $C_{\mathrm{p}}(x|y)$. Since the P-frame coding indicates the differences between two frames, which is essential in finding the (dis)similarity between them, it is encoded with maximum resolution, i.e., minimum quantization scale, which is one in MPEG-1 (the quantization scale for I-frame does not have any effect, as I-frame coding is bypassed). Second, it is noticed that because the second image/patch is compressed in reference to the first one, $C_{\mathrm{p}}(x|y)$ (also $C(x|y)$) is not symmetric. However, if both x and y are from the same distribution (class), one expects $C_{\mathrm{p}}(x|y)$ to be close to $C_{\mathrm{p}}(y|x)$ (because x and y are from the same class and it does not make very much difference whether x is compressed in respect to y or y in respect to x), while if x and y are from different distributions (classes), $C_{\mathrm{p}}(x|y)$ and $C_{\mathrm{p}}(y|x)$ should be largely different. Hence, the new measure is proposed as follows

$$d_{\mathrm{N}}(x,y) = \frac{|C_{\mathrm{p}}(x|y) - C_{\mathrm{p}}(y|x)|}{C_{\mathrm{p}}(x|y) + C_{\mathrm{p}}(y|x)}, \tag{5.3}$$

where the absolute of the difference is taken in the numerator to ensure positive distances. $C_{\mathrm{p}}(x|y) + C_{\mathrm{p}}(y|x)$ is used as the normalizing factor. The proposed distance is symmetric and nonnegative.

Although MPEG-1 has also been used in [88] for a dissimilarity measure, our proposed

measure is different in the following aspects. First, our proposed formulation is different from what they have proposed. Their distance measure is defined as follows

$$d_{\mathrm{CK}}(x,y) = \frac{C(x|y) + C(y|x)}{C(x|x) + C(y|y)} - 1, \tag{5.4}$$

where $C(x|y)$ is computed based on both I- and P-frame coding, while in our approach, it is computed solely based on P-frame coding (denoted by $C_{\mathrm{p}}(.|.)$). Second, in (5.4), the compression is maximized by using large quantization scales for both I- and P-frame coding through MPEG-1 external parameters, to prefer compressibility over image quality [88]. In our approach, since P-frame is essential in finding the (dis)similarity between two frames, it is encoded with maximum resolution. Since I-frame coding is not utilized in our approach, it can be bypassed in MPEG-1 coding, which improves the speed of the computation by 40% compared to the computation of the d_{CK} measure. Third, our proposed measure performs properly on both small and large patches, while $d_{\mathrm{CK}}(x,y)$ cannot represent dissimilarity between small patches properly. This is explained more in the next subsection.

5.5.1 Some Illustrative Results on Textures

To better realize how $d_{\mathrm{CK}}(x,y)$ works, the distances are computed among patches of 17×17, 33×33, 65×65, and 129×129 extracted from two texture images of Brodatz, i.e., D4 (Figure 5.1a) and D5 (Figure 5.1f), as shown in Figure 5.1b-5.1e. As can be seen, the distances computed (300 patches per class) among patches are normalized to the interval of $[0,1]$ to ease the comparison, and displayed using a color code. one expects to see smaller distances among patches extracted from the same class, i.e., in $c_i - c_i, i = 1,2$ areas, and larger distances among the patches extracted from two different classes, i.e., in $c_i - c_j, i, j = 1,2$ & $i \neq j$ areas (see Figure 5.1b as a reference). However, except for the large patch size of 129×129, this behavior cannot be observed in Figure 5.1b-5.1e. This problem can be also seen for any other texture pair, the main reason for which is explained next.

The major problem with $d_{\mathrm{CK}}(x,y)$ defined in (5.4) is that it compresses the concatenated patches based on both I- and P-frames. This is while only P-frame coding is based

on the (dis)similarity of patches and I-frame coding is performed using DCT solely based on the frequency contents of a patch/image. This means that for small patch sizes, where the compression based on P-frame is still limited (due to the small search region) compared to I-frame coding, the distances are mainly dominated by I-frame coding, i.e., frequency contents and distributions of the first frame. Hence, the patches from the texture class that have low frequency contents usually show lower distances (in this case D5; one can investigate this by taking the Fourier transform of both textures and looking at their spectrum). However, in this example, due to D4's greater homogeneity, lower distances are expected among the patches extracted from D4, i.e., in region $c_1 - c_1$.

Figure 5.1g-5.1j shows the distances computed using our proposed measure among the same patches used for d_{CK}, to illustrate the effectiveness of the proposed distance on finding the (dis)similarities among texture pairs. It can be seen that the distances are consistently small among the patches of the same class for all patch sizes, and also that the distances among the patches extracted from D4, which is a more homogeneous texture than D5, are smaller. This behavior can be consistently observed on other texture pairs; two more examples are shown in Figure 5.2.

5.6 Kernelized Supervised Dictionary Learning Using Data-Dependent Kernels

Here, the benefit of using data-dependent kernels, such as kernels computed using the proposed compression-based measure, in the application of texture classification is shown.

Texture provides important information in various fields of image analysis and computer vision. It has been used in many different problems including texture classification, texture segmentation, texture synthesis, material recognition, 3D shape reconstruction, color-texture analysis, appearance modeling, and indexing [81, 95–97].

Texture analysis is an important area of research, judging by the vast literature in this field and its numerous applications in different disciplines [81]. Some of these applications are in remote sensing [98], medical imaging [25–27, 99–101], industrial applications like

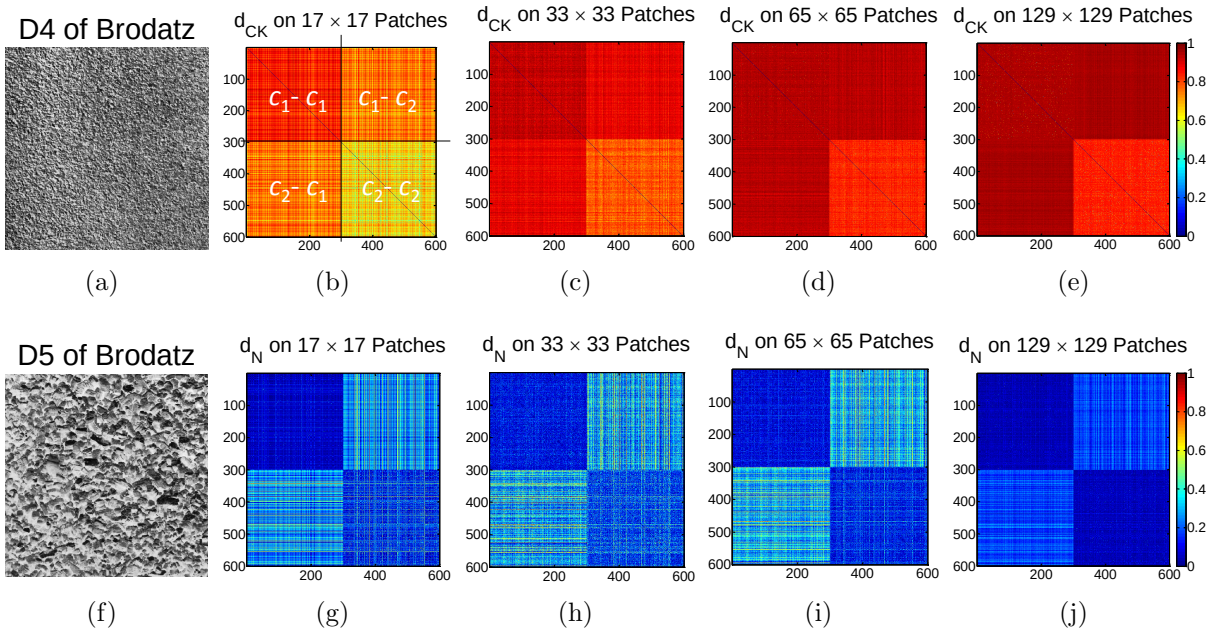


Figure 5.1: The distances computed on patches extracted from (a) D4 and (f) D5 of Brodatz album. (b) to (e) distances computed on various patch sizes as indicated in the figures using d_{CK} and (g) to (j) using proposed measure (d_{N}).

fabric defect detection [82], and object recognition [102].

Texture images can be divided into two broad types: *stationary*, containing only one texture type per image, and *nonstationary* consisting of more than one texture type per image [95]. The main application domain on stationary texture images is supervised classification of each texture image into one class, whereas for nonstationary texture images, there are two main application domains [82,95]. First, unsupervised texture segmentation, which partitions the texture image into disjoint regions of uniform texture. Second, pixel-based texture classification, which is similar to texture segmentation in the sense that the given texture image is segmented to uniform texture regions. The difference, however, is that in pixel classification, the segmentation is performed using supervised techniques [82].

In the following subsections, it is demonstrated how the data-dependent kernels can be incorporated into the kernelized SDL to significantly improve the classification of stationary

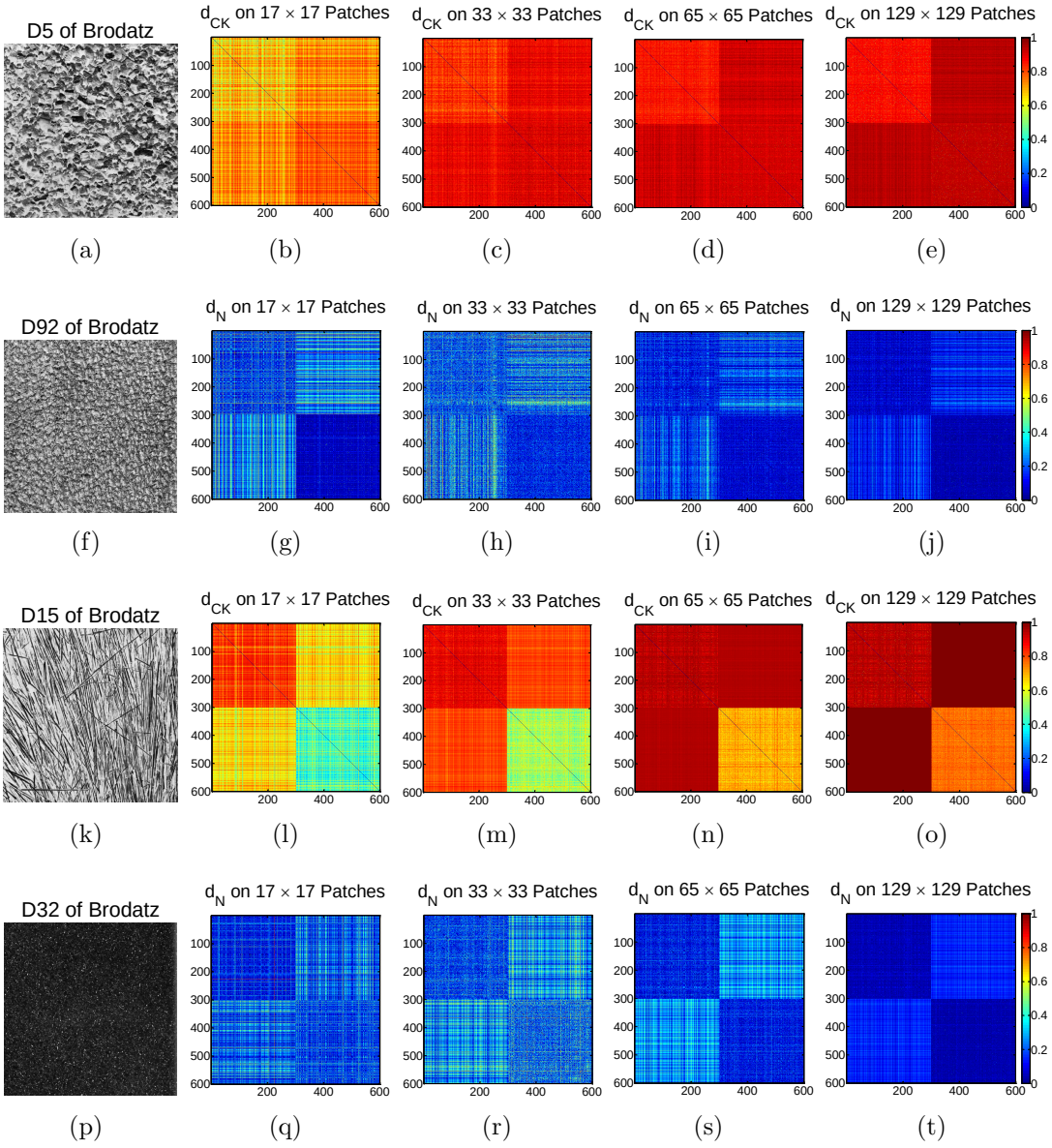


Figure 5.2: The distances computed on patches extracted from two more texture pairs from the Brodatz album. The first two rows display the texture pair (a) D5 and (b) D92, and the corresponding distances computed using (b)-(e) d_{CK} and (g)-(j) proposed measure (d_{N}). Rows 3 and 4 display similarly these distances for texture pair (k) D15 and (p) D32, where (l)-(o) are distances based on d_{CK} and (q)-(t) are those based on proposed measure (d_{N}).

and nonstationary texture images.

5.6.1 Texture Classification on Stationary Images

In this subsection, classification is performed on patches extracted from stationary texture images. To show the gain obtained from incorporating data-dependent kernels into the proposed kernelized SDL (Chapter 3), the results are provided for texture classification with and without kernels using the proposed SDL, and also compared to the results published in [16], i.e., two supervised dictionary learning approaches called SDL-G BL (G for generative and BL for bilinear model) and SDL-D BL (D for discriminative). To ease the comparison, the same data as in [16] is used, i.e., classification on texture pair of D5 and D92 from the Brodatz album, shown in Figure 5.3. Also the same as [16], 300 patches are randomly extracted from the left half of each texture image for training and 300 patches from the right half for testing. This is to ensure that there is no overlap among the patches used in the training and test sets.

An RBF kernel and also two data-dependent compression-based kernels as described in previous sections, i.e., d_{CK} [88] and d_{N}, proposed in Section 5.5 are used as the kernel for the proposed kernelized SDL. As was explained in previous sections, both d_{CK} and d_{N} deploys MPEG-1 as the compressor for the computation of the compression-based dissimilarity measure. However, in contrast to d_{CK} that only performs well at large patch sizes, d_{N} performs well on both small and large patch sizes (refer to the end of Section 5.5 for a comparison between d_{N} and d_{CK}).

Table 5.1 provides the results of classification using the proposed SDL with and without kernels. It also compares the results with k-means as an unsupervised approach to compute the dictionary, and with the results published in [16] for the same number of patches (300) and the same dictionary size, i.e., 64. The sparsity of the coefficients, i.e., the number of nonzero coefficients, are also provided in this table (this is not reported for SDL-G BL and SDL-D BL in [16]). As can be seen, using a compression-based data-dependent kernel based on d_{N} dramatically improves the results. The classification error is even lower than the one obtained by the SDL-D BL approach using 30000 patches for training, which yields the best results on this data with the classification error = 14.26% in [16].

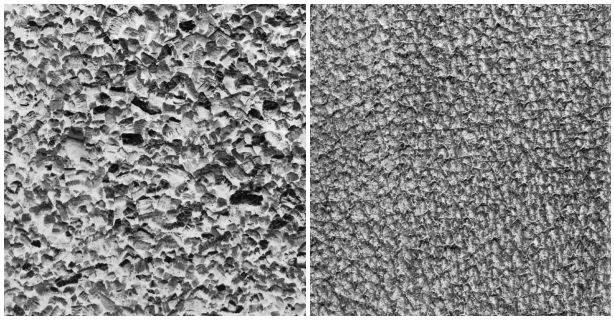


Figure 5.3: Texture images of D5 and D92 from Brodatz album.

Moreover, as the sparsity of the coefficients indicates, the proposed approach with data-dependent kernel d_{N} deploys the smallest number of dictionary atoms in the reconstruction of the signal, i.e., benefits the most from the sparse representation, as it uses almost half of the dictionary elements compared to other approaches. This has a great impact on the computation load of the classification task, especially in the stage of training and testing of the classifier. Our experiments show (not reported in Table 5.1) that by using a slightly larger regularization parameter λ in the *lasso* such that the reconstruction error is within one standard deviation of the minimum, the sparsity of coefficients can be even greater. That is, the average number of nonzero coefficients can be reduced to about 5% of the total number of coefficients, without compromising the classification error. The classification error is 9.90 ± 1.43 in this case, which is even slightly better than what is reported in Table 5.1.

5.6.2 Pixel-Based Texture Classification on Nonstationary Images

In this subsection, the effectiveness of the proposed measure as a data-dependent kernel in the kernelized SDL is shown in the application of supervised pixel-based classification on nonstationary texture images. In this application, there is a trade-off between the patch sizes at smooth areas and on the borders. While a large patch size at the uniform texture areas improves the performance of classification (as more information is included to identify

Table 5.1: Classification error and the number of nonzero coefficients on the test set for texture pair D5-D92 of Brodatz album. The dictionary size is 64. Using data-dependent kernels and the proposed kernelized SDL can significantly improve the results.

Approach		Average No. of Nonzero Coefficients		Classification Error (%)
		Train Set	Test Set	
k-means		47.85	48.99	27.75±2.29
Proposed SDL		59.80	59.85	26.43±2.95
Proposed kernel SDL	RBF	62.88	62.51	28.85±1.84
	CK-1 [88]	64	64	26.05±1.07
	d_{N} [35]	**33.46**	**31.53**	**10.03±1.31**
SDL-G BL[a] [16]		-	-	26.34
SDL-D BL[a] [16]		-	-	26.34

[a]The average no. of nonzero coefficients is not provided for this approach in [16].

the textures correctly), small patch sizes are more desired on the borders to prevent mixing textures from two different classes.

Here, the distances are first computed on 200 patches per class with the size of 17×17 extracted from the training images. A linear kernel $k_{\mathrm{tr}} = d_{\mathrm{tr}}.d'_{\mathrm{tr}}$ is then computed, where d_{tr} is the distance matrix computed on the patches extracted from the training set. This kernel is used in Algorithm 2 (given in Section 3.3) to learn a dictionary of size 64, and subsequently the coefficients for the training set. Then the patches of the same size are extracted from the test image, and the distances among these patches and the training patches are computed using a distance function. A linear kernel is computed subsequently using $k_{\mathrm{ts}} = d_{\mathrm{ts}}.d'_{\mathrm{tr}}$, where d_{ts} is the computed distances from the test to training patches. This kernel, along with the dictionary learned in the previous step, is used in Algorithm 2 to learn the coefficients for the test set. A linear SVM, whose optimal cost function (C^*) is tuned in a 5-fold cross-validation on the training set, is used as the classifier.

The data used is the same as that used in [82]. It is comprised of some texture composites from Brodatz and some outdoor images. The test images are shown in the first column of Figure 5.4. The results are compared to two other distance measures using the

Table 5.2: The classification rate (%) compared among the proposed method and other distance- or feature-based approaches. The results on LBP (local binary pattern) and MeasTex (Gabor, 5NN) methods are based on what is reported in [82] for the same images.

Approach	Test Images				Average
	Figure 5.4a	Figure 5.4f	Figure 5.4k	Figure 5.4p	
Proposed (d_{N})	**88.5**	**82.2**	**74.6**	71.3	**79.2**
d_{CK}	83.4	74.2	72.6	**73.8**	76.0
d_{NCD}	82.7	74.4	74.2	70.6	75.5
$LBP^{\mathrm{riu2}}_{8,1}$ [82]	85.4	77.5	69.4	37.9	67.6
MeasTex (Gabor, 5NN) [82]	83.7	70.5	68.5	55.1	69.5

d_{CK} and NCD approach, and also to two feature-based approaches published in [82] that yield the best results on these texture images, i.e., local binary pattern ($\mathrm{LBP}^{\mathrm{riu2}}_{8,1}$) and MeasTex (Gabor, 5NN) (refer to Table 3 of [82]). To get rid of the speckle-noise type in final classification, just as in [82], a median filter with the same size as the patch sizes (17×17 in this case) is applied to the final classified pixels. The results are shown quantitatively in Table 5.2, and qualitatively in Figure 5.4. As can be seen, our results are better than other distance-based approaches as well as those reported in [82].

5.7 Summary

In this chapter, a novel compression-based distance measure using an MPEG-1 encoder is proposed that takes into account the spatial locality and connectivity of pixels in images. The proposed measure computes distances based on P-frame coding, and can properly find the distances on both small and large patch sizes. By bypassing the I-frame coding, which is not necessary in the computation of distances in the proposed approach, the performance in terms of speed is improved by 40% compared to the closest rival, i.e., the d_{CK}, which has also been designed based on an MPEG-1 compressor [88]. The effectiveness of the proposed measure was shown in the application of texture classification to stationary and

nonstationary texture images. It was shown that embedding the proposed measure as a data-dependent kernel into the kernelized version of the proposed SDL can significantly improve the results of texture classification of stationary images compared with another state-of-the-art SDL approaches, such as those published in [16]. It was also shown that the proposed measure outperforms other compression-based measures, and also feature-based approaches, in the application of pixel-based texture classification.

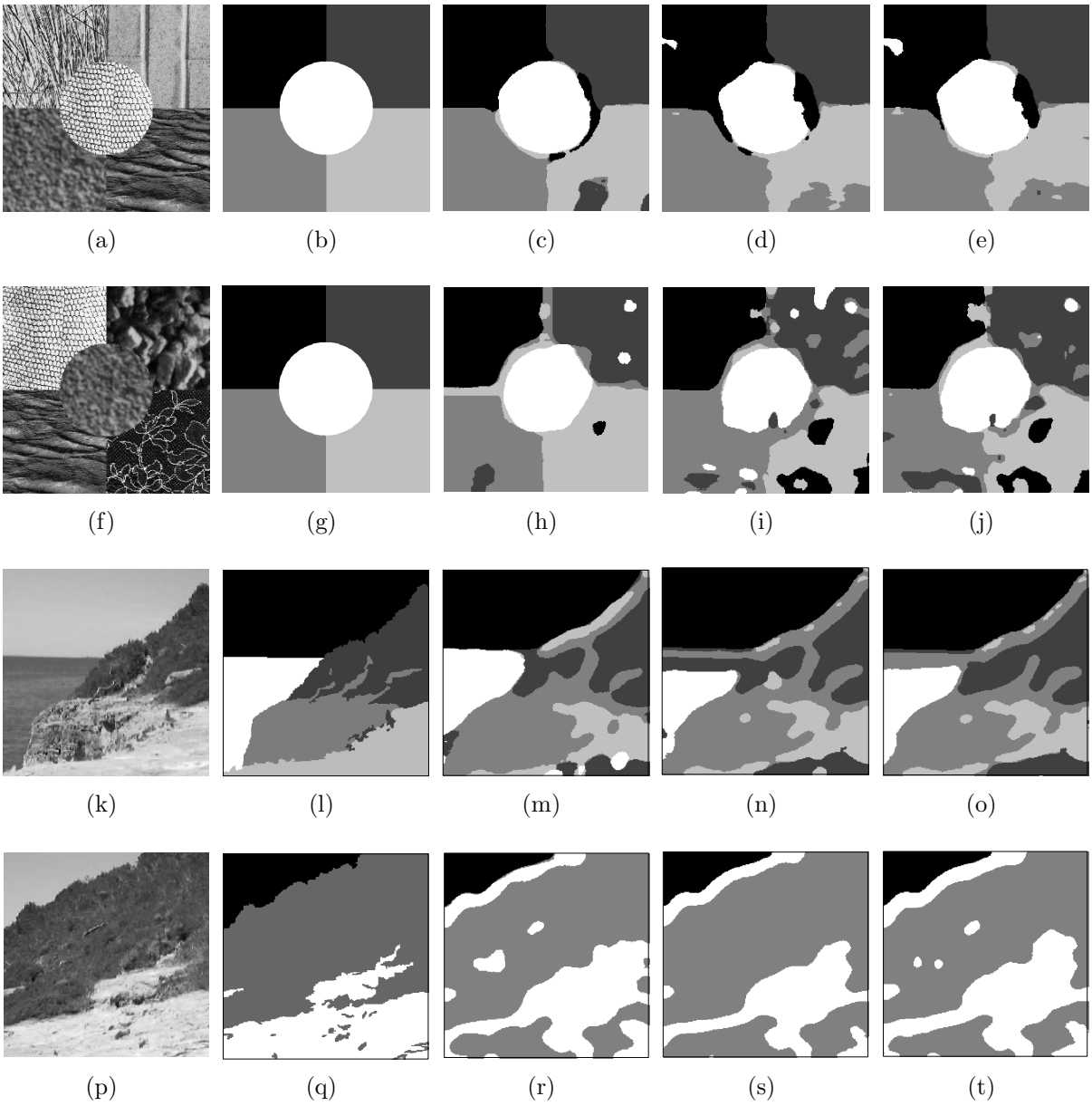


Figure 5.4: The results of supervised pixel-based texture classification on Brodatz and outdoor images. (a, f, k, and p) test images; (b, g, l, and q) ground truth; (c, h, m, and r) proposed measure d_{N}; (d, i, n, and s) d_{CK}; (e, j, o, and t) NCD.

Chapter 6

Extension of the Proposed SDL to Multiview Representations

In previous chapters, a supervised dictionary learning (SDL) technique and its kernelized counterpart were developed based on the Hilbert Schmidt independence criterion (HSIC). It was also shown how embedding data-dependent kernels into the proposed kernelized SDL can significantly improve the performance of the classification systems in subtle problems. Thus far, however, our focus has been on single-view problems. In this chapter, the proposed SDL is extended to multiview signals, and it will be shown how it works in the application of emotion recognition.

6.1 Introduction

There are instances where the data in a dataset is represented in multiple views [103]. This can be due to the availability of several feature sets for the same data such as representation of a document in several languages [104], representation of webpages by both their text and hyperlinks, etc., or due to the availability of information from several modalities, e.g., biometric information for the purpose of authentication may come from fingerprints, iris, face, and so on. Although single-view representation might be sufficient in a ma-

chine learning task for the analysis of the data, complementary information provided from multiple views usually facilitates the improvement of the learning process.

Multiview analysis has been used in many applications, such as emotion recognition by using both audio and visual information [105, 106], biometry and authentication [107, 108], and multilingual document analysis [104]. It has also been used in many different learning paradigms such as in: clustering (unsupervised learning) [109]; semisupervised learning, mainly using techniques based on co-training [103] in applications such as domain adaptation [110]; supervised learning, based on so-called multiple kernel learning (MKL) techniques; also in dimensionality reduction [111].

In classification tasks, learning from multiview data is usually performed either by combining at the feature level or at the decision level. In the former, features from different views are fused, possibly after some normalization, to make a single fused feature set, which is then submitted to a classifier. In the latter case, the feature subset from each single view is submitted to a classifier, and then decisions of these classifiers are combined [112–114].

In the next section, the formulation is provided for the extension of the proposed SDL to multiview representation.

6.2 Multiview Supervised Dictionary Learning

In this section, the formulation for two-view supervised dictionary learning is provided; the extension to more than two views is straightforward. The main assumption is that both views agree on the class labels of all instances in the training set. Let $\mathbf{X}^{(v)} \in \mathbb{R}^{p_1 \times n}$ and $\mathbf{X}^{(w)} \in \mathbb{R}^{p_2 \times n}$ be two views/representations of n training samples with the dimensionalities of p_1 and p_2, respectively. Having these two representations, the main question is how to perform the learning task using the proposed SDL provided in Algorithm 1. There are two approaches, as follows:

Method 1: One approach is to fuse the feature sets from the two views to obtain $\mathbf{X} = \begin{bmatrix} \mathbf{X}^{(v)} \\ \mathbf{X}^{(w)} \end{bmatrix}$, where $\mathbf{X} \in \mathbb{R}^{(p_1+p_2) \times n}$. To learn the supervised dictionary, one needs to use

the optimization problem in (3.11), which is repeated below for the ease of reference.

$$\begin{aligned} \max_{\mathbf{U}} \quad & \text{tr}(\mathbf{U}^\top \mathbf{XHLHX}^\top \mathbf{U}), \\ \text{s.t.} \quad & \mathbf{U}^\top \mathbf{U} = \mathbf{I}. \end{aligned} \tag{6.1}$$

The columns of $\mathbf{U}$, which are the eigenvectors of $\mathbf{\Phi} = \mathbf{XHLHX}^\top$, construct the dictionary $\mathbf{D} \in \mathbb{R}^{(p_1+p_2)\times k}$, where k is the number of dictionary atoms. Using the *lasso* given in (2.3), one can subsequently compute the sparse coefficients $\boldsymbol{\alpha} \in \mathbb{R}^{k\times n}$. These coefficients are submitted to a classifier such as SVM for training.

Method 2: The alternative approach is to learn one subdictionary from the data samples in each view. In other words, by replacing $\mathbf{X}^{(v)} \in \mathbb{R}^{p_1\times n}$ into (6.1) we have

$$\begin{aligned} \max_{\mathbf{U}^{(v)}} \quad & \text{tr}(\mathbf{U}^{(v)^\top} \mathbf{X}^{(v)} \mathbf{HLHX}^{(v)^\top} \mathbf{U}^{(v)}), \\ \text{s.t.} \quad & \mathbf{U}^{(v)^\top} \mathbf{U}^{(v)} = \mathbf{I}. \end{aligned} \tag{6.2}$$

By computing the top k_1 eigenvectors of $\mathbf{\Phi}^{(v)} = \mathbf{X}^{(v)}\mathbf{HLHX}^{(v)^\top}$, a subdictionary $\mathbf{D}^{(v)} \in \mathbb{R}^{p_1\times k_1}$ is obtained, where k_1 is the size of the subdictionary for this view.

Similarly, another subdictionary $\mathbf{D}^{(w)} \in \mathbb{R}^{p_2\times k_2}$ with the size of k_2 can be computed by replacing $\mathbf{X}^{(w)} \in \mathbb{R}^{p_2\times n}$ into (6.1), i.e.,

$$\begin{aligned} \max_{\mathbf{U}^{(w)}} \quad & \text{tr}(\mathbf{U}^{(w)^\top} \mathbf{X}^{(w)} \mathbf{HLHX}^{(w)^\top} \mathbf{U}^{(w)}), \\ \text{s.t.} \quad & \mathbf{U}^{(w)^\top} \mathbf{U}^{(w)} = \mathbf{I} \end{aligned} \tag{6.3}$$

and computing the top k_2 eigenvectors of $\mathbf{\Phi}^{(w)} = \mathbf{X}^{(w)}\mathbf{HLHX}^{(w)^\top}$. By replacing the data samples of each view and their corresponding subdictionaries computed in previous step in the *lasso* given in (2.3), the corresponding coefficients $\boldsymbol{\alpha}^{(v)} \in \mathbb{R}^{k_1\times n}$ and $\boldsymbol{\alpha}^{(w)} \in \mathbb{R}^{k_2\times n}$ can be computed for each view. Each of these coefficients can be interpreted as the representation of the data samples in the space of the subdictionary of the corresponding view. These coefficients are then fused such that $\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}^{(v)} \\ \boldsymbol{\alpha}^{(w)} \end{bmatrix}$, where $\boldsymbol{\alpha} \in \mathbb{R}^{(k_1+k_2)\times n}$. Fused coefficients $\boldsymbol{\alpha}$ are eventually submitted to a classifier such as SVM for training.

Remark: The approach provided in Method 2 can be considered to be a special case of Method 1. To better realize how these two approaches are related, $\mathbf{U}$ in Method 1 can be considered to be of the special form as follows

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}^{(v)} & 0 \\ 0 & \mathbf{U}^{(w)} \end{bmatrix}. \tag{6.4}$$

Considering this form of $\mathbf{U}$, it is easy to show:

(a) The constraint given in (6.1) is equivalent to two constraints given in (6.2) and (6.3):

$$\begin{aligned} & \mathbf{U}^\top \mathbf{U} = \mathbf{I}_k \\ \Rightarrow \quad & \begin{bmatrix} \mathbf{U}^{(v)} & 0 \\ 0 & \mathbf{U}^{(w)} \end{bmatrix}^\top \begin{bmatrix} \mathbf{U}^{(v)} & 0 \\ 0 & \mathbf{U}^{(w)} \end{bmatrix} = \mathbf{I}_k \\ \Rightarrow \quad & \begin{bmatrix} \mathbf{U}^{(v)^\top} \mathbf{U}^{(v)} & 0 \\ 0 & \mathbf{U}^{(w)^\top} \mathbf{U}^{(w)} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{k_1} & 0 \\ 0 & \mathbf{I}_{k_2} \end{bmatrix}, \end{aligned} \tag{6.5}$$

where $\mathbf{I_k}$ is a $k \times k$ identity matrix and $k = k_1 + k_2$. From the last equality in (6.5), it is easy to conclude the constraints given in (6.2) and (6.3), i.e., $\mathbf{U}^{(v)^\top} \mathbf{U}^{(v)} = \mathbf{I}_{k_1}$ and $\mathbf{U}^{(w)^\top} \mathbf{U}^{(w)} = \mathbf{I}_{k_2}$, where the dimensionality of the identity matrices is explicitly shown to prevent confusion. This, consequently means that the dictionaries learned by the two approaches are the same for this special form of $\mathbf{U}$ given in (6.4).

(b) Also, the coefficients $\boldsymbol{\alpha}$ obtained from Method 1 will be equivalent to the coefficients $\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}^{(v)} \\ \boldsymbol{\alpha}^{(w)} \end{bmatrix}$ computed using Method 2. This can be shown by using the *lasso* given in (2.3), the special form of $\mathbf{U}$ given in (6.4), and by recalling that $\mathbf{X} = \begin{bmatrix} \mathbf{X}^{(v)} \\ \mathbf{X}^{(w)} \end{bmatrix}$ as follows[1]:

$$\begin{aligned} \|\mathbf{X} - \mathbf{U}\boldsymbol{\alpha}\|_{\mathrm{F}}^2 + \lambda \|\boldsymbol{\alpha}\|_1 &= \left\| \begin{bmatrix} \mathbf{X}^{(v)} \\ \mathbf{X}^{(w)} \end{bmatrix} - \begin{bmatrix} \mathbf{U}^{(v)} & 0 \\ 0 & \mathbf{U}^{(w)} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}^{(v)} \\ \boldsymbol{\alpha}^{(w)} \end{bmatrix} \right\|_{\mathrm{F}}^2 + \lambda \left\| \begin{bmatrix} \boldsymbol{\alpha}^{(v)} \\ \boldsymbol{\alpha}^{(w)} \end{bmatrix} \right\|_1 \\ &= \left\| \begin{bmatrix} \mathbf{X}^{(v)} \\ \mathbf{X}^{(w)} \end{bmatrix} - \begin{bmatrix} \mathbf{U}^{(v)} \boldsymbol{\alpha}^{(v)} \\ \mathbf{U}^{(w)} \boldsymbol{\alpha}^{(w)} \end{bmatrix} \right\|_{\mathrm{F}}^2 + \lambda \left\| \begin{bmatrix} \boldsymbol{\alpha}^{(v)} \\ \boldsymbol{\alpha}^{(w)} \end{bmatrix} \right\|_1. \end{aligned} \tag{6.6}$$

[1] $\mathbf{U}$ is used instead of $\mathbf{D}$ in the *lasso* as the dictionary elements are the columns of $\mathbf{U}$. See also footnote 2 in Chapter 2.

Algorithm 3 Multiview Supervised Dictionary Learning-Method 1 (MV1)

Input: Training data at multiple views, $\mathbf{X}_{\text{tr}}^{(v)}, v = 1, ..., \text{V}$, test data at multiple views, $\mathbf{X}_{\text{ts}}^{(v)}, v = 1, ..., \text{V}$, kernel matrix of labels $\mathbf{L}$, training data size, n, size of dictionary, k.

Output: Dictionary, $\mathbf{D}$, coefficients for training and test data, $\boldsymbol{\alpha}_{\text{tr}}$ and $\boldsymbol{\alpha}_{\text{ts}}$.

1: $\mathbf{X}_{\text{tr}} = \begin{bmatrix} \mathbf{X}_{\text{tr}}^{(1)} \\ \vdots \\ \mathbf{X}_{\text{tr}}^{(\text{V})} \end{bmatrix}$

2: $\mathbf{X}_{\text{ts}} = \begin{bmatrix} \mathbf{X}_{\text{ts}}^{(1)} \\ \vdots \\ \mathbf{X}_{\text{ts}}^{(\text{V})} \end{bmatrix}$

3: $\mathbf{H} \leftarrow \mathbf{I} - n^{-1}\mathbf{e}\mathbf{e}^{\top}$

4: $\boldsymbol{\Phi} \leftarrow \mathbf{X}_{\text{tr}}\mathbf{H}\mathbf{L}\mathbf{H}\mathbf{X}_{\text{tr}}^{\top}$

5: **Compute Dictionary:** $\mathbf{D} \leftarrow$ eigenvectors of $\boldsymbol{\Phi}$ corresponding to top k eigenvalues

6: **Compute Training Coefficients:** Replace $\mathbf{X}$ with $\mathbf{X}_{\text{tr}}$ in (2.3), use (2.3) to compute $\boldsymbol{\alpha}_{\text{tr}}$ given $\mathbf{D}$

7: **Compute Test Coefficients:** Replace $\mathbf{X}$ with $\mathbf{X}_{\text{ts}}$ in (2.3), use (2.3) to compute $\boldsymbol{\alpha}_{\text{ts}}$ given $\mathbf{D}$

The bottom line of (6.6) is effectively consisting of two *lasso* formulations, i.e., $\|\mathbf{X}^{(v)} - \mathbf{U}^{(v)}\boldsymbol{\alpha}^{(v)}\|_{\text{F}}^2 + \lambda\|\boldsymbol{\alpha}^{(v)}\|_1$ for view v and $\|\mathbf{X}^{(v)} - \mathbf{U}^{(w)}\boldsymbol{\alpha}^{(w)}\|_{\text{F}}^2 + \lambda\|\boldsymbol{\alpha}^{(w)}\|_1$ for view w. This shows that for the special form of $\mathbf{U}$ given in (6.4), the coefficients computed using Method 1 are the same as what are computed using Method 2.

In summary, it can be concluded that by adding an additional constraint on $\mathbf{U}$, Methods 1 and 2 yield the same results, i.e., the same dictionary and coefficients. This additional constraint on $\mathbf{U}$ is that it should be of the form given in (6.4). This special form, effectively, decouples the computation of the dictionary and coefficients over two views. Algorithms 3 and 4 provides the computation steps for the two multiview approaches proposed above. In the following sections, the relative performance of these two multiview approaches is shown in the applications of emotion recognition.

Algorithm 4 Multiview Supervised Dictionary Learning-Method 2 (MV2)

Input: Training data at multiple views, $\mathbf{X}_{\text{tr}}^{(v)}, v = 1, ..., \text{V}$, test data at multiple views, $\mathbf{X}_{\text{ts}}^{(v)}, v = 1, ..., \text{V}$, kernel matrix of labels $\mathbf{L}$, training data size, n, size of dictionary, k.

Output: Dictionary, $\mathbf{D}$, coefficients for training and test data, $\boldsymbol{\alpha}_{\text{tr}}$ and $\boldsymbol{\alpha}_{\text{ts}}$.

1: $\mathbf{H} \leftarrow \mathbf{I} - n^{-1}\mathbf{e}\mathbf{e}^{\top}$

2: **for** $v = 1 \rightarrow \text{V}$ **do**

a: $\boldsymbol{\Phi}^{(v)} \leftarrow \mathbf{X}_{\text{tr}}^{(v)}\mathbf{H}\mathbf{L}\mathbf{H}\mathbf{X}_{\text{tr}}^{(v)^{\top}}$

b: $\mathbf{D}^{(v)} \leftarrow$ eigenvectors of $\boldsymbol{\Phi}^{(v)}$ corresponding to top k eigenvalues

c: Replace $\mathbf{X}$ with $\mathbf{X}_{\text{tr}}^{(v)}$ in (2.3), use (2.3) to compute $\boldsymbol{\alpha}_{\text{tr}}^{(v)}$ given $\mathbf{D}^{(v)}$

d: Replace $\mathbf{X}$ with $\mathbf{X}_{\text{ts}}^{(v)}$ in (2.3), use (2.3) to compute $\boldsymbol{\alpha}_{\text{ts}}^{(v)}$ given $\mathbf{D}^{(v)}$

3: **end for**

4: **Compute Dictionary:** $\mathbf{D} \leftarrow \begin{bmatrix} \mathbf{D}^{(1)} & 0 & \cdots & 0 \\ 0 & \mathbf{D}^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{D}^{(\text{V})} \end{bmatrix}$

5: **Compute Training Coefficients:** $\boldsymbol{\alpha}_{\text{tr}} \leftarrow \begin{bmatrix} \boldsymbol{\alpha}_{\text{tr}}^{(1)} \\ \vdots \\ \boldsymbol{\alpha}_{\text{tr}}^{(\text{V})} \end{bmatrix}$

6: **Compute Test Coefficients:** $\boldsymbol{\alpha}_{\text{ts}} \leftarrow \begin{bmatrix} \boldsymbol{\alpha}_{\text{ts}}^{(1)} \\ \vdots \\ \boldsymbol{\alpha}_{\text{ts}}^{(\text{V})} \end{bmatrix}$

6.3 Multiview SDL in Facial Expression Recognition

Emotion/expression recognition is an emerging field of research due to its wide range of applications, including human-computer interaction, security and surveillance, data-driven animation, interactive video, indexing video and images, driver safety, health care, and game/entertainment industries [115–117].

Most of the research in recent years has focused on emotion recognition using voice

and facial expressions. Despite many recent advances in the field, audio/visual expression recognition remains a difficult task, due to the variability and complexity of these expressions. In this section, the focus is on facial expression recognition, and in next, speech emotion recognition is addressed.

Psychophysical and neurophysiological research indicates that there are six basic emotions, which are universal across cultures. The pioneer work by Ekman in this direction specifies these basic emotions as happiness, surprise, disgust, fear, anger, and sadness [118]. The main goal in many research works conducted on facial expression recognition is to categorize the human face to one of these prototypical emotions, along with a neutral facial expression [105, 115, 119–122]. The same 7-class problem is used for the classification of facial expressions in this section.

There have recently been much research published in the area of facial expression recognition (FER), each of which addresses one or several aspects of this demanding research area. For example, which feature representation works better in FER was empirically investigated in [115]. They concluded that local binary patterns (LBP) along with SVM perform slightly better than Gabor-wavelet-based SVMs. Further, LBP is much more efficient than Gabor-wavelets in terms of speed and memory usage. There are other papers that address the dimensionality reduction in FER. For example, Liu *et al.* propose a dimensionality reduction based on tensor rank-one decomposition for multidimensional data in FER [122]. 3D FER has been addressed in [119,120]. A survey of the field can be found in [117].

Extracting significant and discriminative facial features to represent the face images is a vital step in the design of a successful FER system. The facial features can be broadly divided into two categories: geometric [123] and appearance-based features [105,115]. Geometric features are computed by extracting the shape and location of the most important components of the face, such as mouth, eyes, eyebrows, and nose. Some parameters may subsequently be measured on these prominent components to represent the face emotion [123, 124]. Appearance-based features, on the other hand, rely mainly on deploying predefined operators such as filter banks to extract features from face images. The most common appearance-based features are Gabor-wavelet features and local binary patterns (LBPs). Although many researchers use features extracted by applying Gabor-wavelet filter

banks to face images for FER, the computation of these features are time- and memory-intensive [115]. LBP features were originally proposed for texture analysis as predefined operators that could efficiently extract important textural characteristics [125]. Ahonen *et al.* were the first who proposed to use them as descriptors in face recognition [126]. Computation of LBP features is much faster than Gabor-wavelets in the application of FER, as shown in [115].

Since geometric and appearance-based features rely on two different procedures for the extraction of facial features, they can potentially provide complementary information on facial expressions. Hence, our conjecture is that deploying both feature sets can improve the performance of a FER system. Among appearance-based features, LBP has been selected due to its lower computational cost. For geometric features, facial measure features (FMF), recently proposed in [123], has been adopted. The proposed SDL and its multiview extensions are used on some face datasets to show the effectiveness of combining these feature sets in an FER system.

6.3.1 Datasets

In this subsection, a brief overview is provided for the face expression datasets used in this research. Table 6.1 summarizes the details of the datasets as used in our experiments. Figure 6.1 shows some sample images from CK+ and JAFFE datasets at various emotion states, while Figure 6.2 displays typical successive frames from several subjects in the VAM dataset.

Extended Cohn-Kanade (CK+) dataset

CK+ dataset [127] is the extended version of the original Cohn-Kanade dataset [128]. CK+ was recorded on 210 adults in ages between 18 and 50 years. Among the participants, 69% were female, 81% Euro-American, 13% Afro-American, and 6% other groups. The participants were requested to perform twenty-three different facial expressions including six basic emotions. The images were recorded in the resolution of 640×490 or 640×480

(a)

(b)

Figure 6.1: Sample images from six face emotion classes (i.e., Anger, Disgust, Fear, Happiness, Sadness, and Surprise) along with the neutral state for (a) CK+ and (b) JAFFE datasets.

pixels with 8-bit gray level or 24-bit color scales. The full description of the dataset is provided in [127].

In our experiments, the face data of 109 sequences were selected. The only criterion for this selection was that the facial expressions can be assigned to one of the six basic emotions. In each sequence, the neutral face, along with the last three frames at the peak of emotion expression, were selected. In total, 1254 frames were selected with 135, 177, 75, 207, 84, 249, and 327 frames/samples in the Anger, Disgust, Fear, Happiness, Sadness, Surprise, and Neutral classes, respectively.

Figure 6.2: Sample frames from four subjects of Vera Am Mittag (VAM) dataset. Seven successive frames from each subject are shown.

Table 6.1: Face emotion datasets used in the experiments.

Dataset	Samples	Subjects	Samples per Class						
			Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral
CK+[a]	1254	118	135	177	75	207	84	249	327
JAFFE[b]	213	10	30	29	32	31	31	30	30
VAM[c]	1320	20	326	29	11	345	143	149	317

[a] http://www.pitt.edu/~jeffcohn/CKandCK+.htm
[b] http://www.kasrl.org/jaffe.html
[c] http://emotion-research.net

The Japanese female facial expression (JAFFE) dataset

The JAFFE dataset [129, 130] is a facial expression database collected from 10 Japanese females. It consists of 213 images at six basic emotions plus neutral state. The images are available at the resolution of 256×256 pixels.

Vera Am Mittag (VAM) dataset

VAM is an authentic spontaneous audio-visual database from real-life conversations, which were recorded from the German TV talk show "Vera am Mittag", in which guests mostly talk about their personal issues in a spontaneous, affective and unscripted manner [131]. The images are extracted from twenty speakers with a total number of 1867 images. A subset of the VAM dataset has been selected, for which the prototypical emotions and geometric features are available. This subset contains 1320 images, which are used in our experiments. The emotions have been ranked by eight to thirty-four evaluators, and majority voting was used to assign the class label to each face image. There are 326, 29, 11, 345, 143, 149, and 317, images in the Anger, Disgust, Fear, Happiness, Sadness, Surprise, and Neutral classes, respectively. Since the number of data samples in classes Disgust and Fear is very small, the classification has been only performed on the 5-class problem excluding these two classes.

6.3.2 Facial Features

As mentioned earlier in this section, two sets of facial representations are extracted for face images. These are local binary patterns (LBP), which is an appearance-based representation, and facial measure features (FMF), which is a geometric-based feature set.

Local binary patterns (LBP)

LBP features are extracted from face images in this work using the approach described in [115]. The face region in each frame is first detected using the real time face detection approach by Viola and Jones [132]. The resulting face regions are then normalized as follows: the coordinates of the two eyes are identified, then the size normalization is done by resizing the image with distance between the eyes of 55 pixels. Afterwards, the whole face image is cropped to the size of 150×110, relative to the eyes' position. The resulting face images are then divided into 7×6 subregions. Finally, the LBP descriptors are extracted for each region, and the histograms are mapped into uniform patterns in an (8,2) neighborhood. The final feature vector is of size 2,478 (7×6×59) for each face.

Facial measure features (FMF)

FMF features are proposed in [123] as geometric features and extracted from three regions of face, i.e., eyes, nose, and mouth. In total, sixteen features are extracted, including seven features from eyes, six from mouth, and three from the nose region. The interested reader can refer to [123] for more details on how these features are computed.

6.3.3 Implementation Details

The features extracted by LBP and FMF are used as views v and w for facial expressions, respectively. In other words, the two views are: $\mathbf{X}^{(v)} \in \mathbb{R}^{2478\times n}$ and $\mathbf{X}^{(w)} \in \mathbb{R}^{16\times n}$.

The implementation of the supervised dictionary learning in single and multiview is similar to that explained in Section 4.1. In all the experiments, the data is normalized such that each feature is in the range of [0,1]. The evaluation of the system is performed in person-independent setting, meaning that the training and test sets are disjoint at subject level, not sample level. To this end, 10-fold cross-validation has been performed on CK+, where the single fold in the test set do not share the face images from the subjects in the folds belonging to the training set to guarantee person-independent evaluation. For the JAFFE and VAM datasets, as there are only ten and twenty subjects, respectively, leave-one-subject-out is computationally feasible and hence, was adopted. Since the labels of all face images in the dataset are eventually predicted in a cross-validation system, the predicted labels are compared with the ground-truth and the performance of the facial expression recognition systems is reported using classification accuracy, where classification accuracy is $\mathrm{A} = 1 - \mathrm{E}$ and E is the classification error defined in (4.1). For the CK+ and VAM datasets, which are highly imbalanced, balanced classification accuracy $\mathrm{BA} = 1 - \mathrm{BE}$ is also reported, where BE is balanced classification error as defined in (4.2).

An SVM with RBF kernel is used for the classification of face emotions. The optimal parameters of the SVM, i.e., the optimal kernel width γ^* and trade-off parameter C^*, are tuned on the training set in a 5-fold cross-validation. This tuning is also performed in a person-independent setting to get more realistic optimal parameters. It should be emphasized here that tuning SVM parameters by 5-fold cross-validation at sample level

instead of subject level does not violate the person-independent setting in our experiments. However, since the tuned SVM is eventually used in a person-independent setting, the optimal parameters would better replicate the actual situation and therefore, lead to a better performance if they are tuned in the same setting.

6.3.4 Results

Here, the accuracy of the classification systems are compared among single-view (SV) and multiviews (MV) approaches. The classification accuracy (A) of the facial expression recognition systems are provided in Table 6.2. For the CK+ and VAM datasets, which are highly imbalanced, balanced classification accuracy (BA) has been also reported in Table 6.3. Table 6.4 provides the average time per fold for the two multiview FER systems, including the time required to learn the dictionary and the coefficients, tune the parameters of the SVM, and its training and testing to compare the speed of the two multiview systems proposed in this chapter. The experiments are performed on varying dictionary sizes. For single-view using LBP features and for multiview approaches, the results are reported at the dictionary sizes of 8, 16, 32, and 64. For single-view using FMF geometric features, the results are reported at dictionary sizes of 2, 4, 8, and 12. This is because there are only 16 features in FMF, and as discussed before for the proposed SDL, the dictionary size can be at most the same size as the dimensionality of the data, which is 16 in this case.

The results reveal several important points. First, in single view, the LBP features are more discriminative on the CK+ and particularly VAM datasets, whereas FMF features are more discriminative on the JAFFE dataset. Second, these two sets of features, one of which is appearance-based (LBP) and the other geometric-based (FMF), indeed provide complementary information such that using both sets in a multiview system benefits from this information and improves the performance of the FER system. Third, the performance of the FER system on the VAM dataset is rather poor compared with the other two datasets. This is mainly because VAM is a spontaneous dataset, as opposed to the other two which contain posed emotions, and hence, it is a much more challenging dataset. To improve the performance of the FER system on VAM, it is needed to extract more discriminative features. Fourthly, overall, the performance of the multiview system based

on Method 2, i.e., the fusion of features at the dictionary spaces, leads to better performance in terms of both accuracy and speed than Method 1, in which the feature sets are fused in the original feature spaces. An interesting fact is that the speed of FER system based on MV2 on average is twice the speed of the one based on MV1.

There are several reasons why MV2 performs better than MV1 in this application. A comparison between the dimensionality of LBP and FMF features shows that the number of features in LBP (2478) is much larger than the number of features in FMF (16). Fusing these two feature sets in the original space may mean that one feature set is dominated by the other and subsequently eclipses the influence of the smaller feature set to some extent. In the MV2 approach, this does not happen, as the dictionary and coefficients are learned in each view separately and then the representations of the data in the new spaces are fused. The dimensionality of feature sets in these new spaces is the same as the number of dictionary atoms that can be controlled to be at the same order. In other words, each of the representations can evenly contribute towards the overall performance of the classification system. Moreover, MV2 provides the flexibility to the designer of the FER system to choose the number of dictionary atoms learned in each view or to tune the optimal number of dictionary atoms in each view by cross-validation on the training set. In MV1, there is no control of the dictionary atoms learned from each view and hence, one cannot determine the ratio of dictionary atoms learned from each view.

To explain why MV2 is faster than MV1, it is noticed that most of the computation time is spent on the tuning and training of the SVM classifier. Also recall from Chapter 3 that the proposed SDL in single-view finds the directions of maximum separation by finding a space where the dependency between the data points and the corresponding labels are maximized based on Hilbert Schmidt independence criterion (HSIC). In MV2, the data in each view is already mapped to a space where it is represented by the directions of maximum separation, and this is done for each view separately. Hence, one should expect that learning in this spaces to be carried out faster.

As a final remark, the proposed multiview SDL introduces a general framework that can be applied to any multiview representation. In this section, it was demonstrated how using the proposed multiview SDL can benefit from the complementary information in different feature sets. By computing more discriminative feature sets from face images, the

Table 6.2: *Classification accuracy* (%) of facial expression recognition system based on single-view (SV) and multiview (MV) supervised dictionary learning approaches discussed in this thesis. MV1 and MV2 are the multiview SDL approaches based on Method 1 and Method 2 as discussed in Section 6.2. The experiments are performed in a person-independent setting. Highlighted entries show the best performance for each dataset.

Dataset	Approach	Dictionary Size[a]			
		8	16	32	64
CK+ 6-Class	SV-LBP	91.05	92.34	92.13	93.42
	SV-FMF	75.62	85.65	86.62	87.38
	MV1	**92.45**	93.10	92.77	93.96
	MV2	92.02	**93.20**	**93.31**	**94.17**
CK+ 7-Class	SV-LBP	85.17	88.68	88.68	89.23
	SV-FMF	67.70	78.07	77.03	75.68
	MV1	86.20	89.31	89.15	90.03
	MV2	**86.44**	**90.43**	**89.79**	**91.07**
JAFFE 7-Class	SV-LBP	52.11	62.44	62.44	64.32
	SV-FMF	46.01	61.97	66.20	68.08
	MV1	56.81	65.73	64.32	66.20
	MV2	**61.97**	**67.14**	**71.83**	**73.24**
VAM 5-Class	SV-LBP	31.02	36.72	37.89	40.16
	SV-FMF	22.81	23.75	28.91	22.34
	MV1	31.64	36.88	39.84	40.94
	MV2	**33.44**	**39.69**	**42.19**	**41.48**

[a]The dictionary sizes in case of single-view with geometric features (SV-FMF) are 2, 4, 8, and 12 instead.

performance of the proposed multiview SDL can be improved.

Table 6.3: *Balanced classification accuracy* (%) of facial expression recognition system for CK+ and VAM datasets, which are highly imbalanced, based on single-view (SV) and multiview (MV) supervised dictionary learning approaches discussed in this thesis. MV1 and MV2 are the multiview SDL approaches based on Method 1 and Method 2 as discussed in Section 6.2. The experiments are performed in a person-independent setting. Highlighted entries show the best performance for each dataset.

Dataset	Approach	Dictionary Size[a]			
		8	16	32	64
CK+ 6-Class	SV-LBP	84.85	86.25	86.12	88.36
	SV-FMF	68.54	78.68	80.64	81.44
	MV1	**87.08**	86.11	87.79	85.20
	MV2	86.62	**88.33**	**88.78**	**89.96**
CK+ 7-Class	SV-LBP	76.45	80.46	80.54	80.99
	SV-FMF	55.25	66.08	65.77	65.86
	MV1	77.94	81.26	80.87	81.59
	MV2	**78.28**	**83.49**	**81.51**	**83.99**
VAM 5-Class	SV-LBP	25.36	30.21	32.45	34.19
	SV-FMF	19.37	19.81	24.19	18.77
	MV1	26.14	30.41	34.11	34.20
	MV2	**28.25**	**32.36**	**35.70**	**35.10**

[a]The dictionary sizes in case of single-view with geometric features (SV-FMF) are 2, 4, 8, and 12 instead.

6.4 Multiview SDL in Speech Emotion Recognition (SER)

Although automatic speech recognition has been around for many years now, it is not always sufficient only to know *what* is said in a conversation, but sometimes we need to know *how* something is said. That is due to the fact that speech can convey much more information than the mere verbal content [133]. Hence, speech emotion recognition

Table 6.4: The *average time* (in seconds) over each fold for the single-view and multiview FER systems including the time required for learning the dictionary and the coefficients, tuning the SVM parameters, and eventually training and testing the SVM using tuned parameters. Highlighted entries show the fastest execution time for each dataset.

Dataset	Approach	Dictionary Size[a]			
		8	16	32	64
CK+ 6-Class	SV-LBP	26.73	43.03	67.46	117.68
	SV-FMF	17.82	**17.60**	**22.51**	**27.95**
	MV1	47.59	69.28	102.06	169.88
	MV2	**16.14**	27.95	48.90	78.41
CK+ 7-Class	SV-LBP	50.88	82.59	129.47	208.98
	SV-FMF	38.95	**39.92**	**48.19**	**56.71**
	MV1	85.13	126.60	185.44	309.17
	MV2	**33.15**	52.79	92.80	175.89
JAFFE 7-Class	SV-LBP	3.90	5.20	7.77	13.32
	SV-FMF	**2.56**	**2.74**	**3.12**	**3.62**
	MV1	5.35	7.00	9.88	16.47
	MV2	3.09	4.10	5.63	8.43
VAM 5-Class	SV-LBP	87.80	104.53	150.35	257.57
	SV-FMF	119.59	100.05	106.04	**97.67**
	MV1	87.71	104.78	150.79	257.06
	MV2	**66.16**	**69.14**	**92.42**	151.27

[a]The dictionary sizes in case of single-view with geometric features (SV-FMF) are 2, 4, 8, and 12 instead.

attempts to identify this information. A natural application of this is to human-computer interaction. That is, to enable computers to adapt to the emotional states of the users, in order to reduce their frustration during interactions [134]. Different modalities (also referred to as social cues) have been used for this purpose, among which only voice cues have led to the discussion of the current section.

Given the speech signal $s[n]$, there are two major stages into a solution for speech emotion recognition: 1) extraction of low-level descriptors (LLD) from speech, and 2) statistical modeling. Extraction of LLDs is essential, as on the one hand, each speech sample does not convey more than the air pressure recorded by the microphone at a very small fraction of time, meaning one is required to calculate some useful measures of speech that have closer relationship with its affective qualities; on the other hand, speech signals are usually of very high dimensions, therefore extracting LLDs also counts as a dimensionality reduction stage. Subsequently, at the second stage, given the LLDs, as the covariates (i.e., $\mathbf{X} \in \mathbb{R}^{p \times N}$)[2], and an affective quality of speech, as the response variable (i.e., $\mathbf{Y} \in [-1, 1]^N$), the idea is to find a mapping between them: $\mathbf{X} \rightarrow \mathbf{Y}$. Later, this mapping will be used to make predictions on the affective qualities of speech samples.

As for the affective qualities, which is denoted by $\mathbf{Y}$, two points of view for representing emotional states have been used: categorical and dimensional. According to the categorical view, emotional states [118,135] can be described using *discrete* emotion categories such as anger or happiness, as discussed in the previous section. On the other hand, a dimensional point of view, also known as primitive-based, suggests the use of some *continuous* lower level attributes, e.g., arousal and valence. Theories behind the dimensional representation claim that the space defined by those representations can subsume all the categorical emotional states [136–138]. Therefore, depending on the choice of affective qualities, the modeling problem can be recognized as either classification or regression, for categorical (discrete) and dimensional (continuous) representations, respectively.

Acoustic LLDs are categorized by their domain of extraction. Those which are interpreted in the time and frequency domains are respectively known as prosodic and spectral LLDs. Among prosodic LLDs, pitch, speaking rate, jitter, shimmer, and harmonics-to-noise ratio (HNR) are frequently applied to emotional speech recognition. On the other hand, Mel frequency cepstrum coefficients (MFCC), formant frequencies, energy in different spectral bands (250-650 Hz and 1-4 kHz), and spectral characteristics such as flux, entropy, variance, skewness, and kurtosis, are among the most commonly-used spectral LLDs [139]. A list of about forty LLDs, including prosodic and spectral, has been recently

[2]In this thesis, n is used as the number of data samples. However, since in this section, n is the notation for discrete time, N is used as the number of data samples instead.

set as a standard [139–141], and it appears that the list has been adopted by the research community [142–147]. Except for a very few studies [146–150], the recent research does not show a major investigation for introduction of new LLDs.

On the statistical modeling side, various models and learning algorithms have been used to tackle the problem at hand. To begin with, all sorts of linear models and learning algorithms for such models have been used: least squares [143], linear discriminant analysis [151], support vector machines (SVM) [142], and support vector regression (SVR) [152]. Furthermore, decision trees [153], specifically random forests [154], and neural networks [155] have been adopted in various studies. K-nearest neighbors (k-NN) has been put into practice in many works [156, 157]. Also, different types of probabilistic models, including naïve Bayes [158], mixture of Gaussian [159], hidden Markov models [160], and conditional random fields [145] have been used in the literature of this problem.

In the remainder of this section, first an overview of the emotional speech database used in our experiments is provided, then our choice of acoustic LLD is descriv=bed, and eventually the experiments and the results are presented.

6.4.1 Datasets

Dozens of emotional speech databases have been collected in the past few years. However, not all of them could attract the attention of the research community. In this subsection, SEMAINE, which has been used in our experiments, is introduced.

SEMAINE

SEMAINE is a database recorded based on the sensitive artificial listener (SAL) interaction scenario [161]. The aim of SAL is to evoke strong emotional responses in a listener by controlling the statements of an operator (the script is predefined in this scenario). For this purpose, four agents are introduced, and a user can decide which operator to talk to at any time. Each of those agents tries to simulate a different personality: Poppy tries to evoke happiness, Obadiah tries to evoke sadness, Spike tries to evoke anger, and Prudence

tries to make people sensible. Therefore, the combination of those decisions is claimed to result in a highly emotional conversation.

Solid SAL [162,163] is a similar scenario to SAL, for which there is no predefined script given to the operators. Instead, they are free to act as one of the four SAL agents at any time. This is done for the sake of a more natural face-to-face conversation. As in the SAL scenario, reading the script or recalling it (in case operators have memorized the script) may not allow such non-verbal interactions.

SEMAINE is recorded using three different scenarios: solid SAL, semi-automatic SAL, and automatic SAL. 150 participants (93 female and 57 male) have taken part in the recordings, and their ages range from 22 to 60 (32.8 ± 11.9).

A major part of the recent studies on emotional speech recognition [139,142–150,164–167] have been conducted relying on the solid-SAL part of the database. For this part of the database, there are 475 minutes of audio and video available for the user's role, and the same amount of time for the operator's role. The total number of sessions is 95. Solid SAL is annotated by five emotional dimensions: valence, activation, power, expectation, and intensity. To briefly explain these dimensions, valence is a measure of indicating pleasantness, activation is a measure of indicating both mental and physical energy, power indicates the extent of control of a subject over a situation, expectation is a measure of evaluating the situational awareness of a subject, and intensity indicates the distance of a subject from rationality, in any direction.

Despite the relatively young age of the database, it has been a target of various studies already. The main reasons for the attraction towards the SEMAINE are first [139] and second [168] audio/visual emotion challenge (AVEC), which have set the solid SAL part of the database as the benchmark. For the sake of these challenges, four out of five dimensions were used, excluding intensity. Our study is conducted based on the fully-continuous sub-challenge (FCSC) of the AVEC 2012. In FCSC, the features are extracted at 0.5 second intervals only during speech [168]. According to the settings of this challenge, three subsets of the database were used for the training, development, and testing purposes. Due to the fact that the labels of the test subset were not released to the public, the experiments are performed based on the other two subsets. That is, the model is trained using the training

set, and then the it is evaluated using the development set. The number of samples in each of the subsets is 10806 and 9312, respectively.

6.4.2 Audio Features

Different acoustic low-level descriptors (LLD), also known as speech features, have been employed for the emotional recognition of speech. In the following, a review of the spectral energy distribution is provided as our choice of LLD followed by the baseline features provided by AVEC 2012 [168].

Spectral energy distribution (SED)

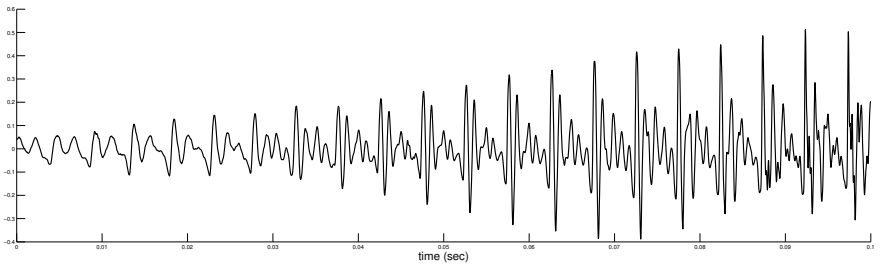
Spectral energy distribution (SED) is comprised of a set of components, where each component represents the relative energy of the signal in a specific band of the spectrum [150,169]. For a speech signal $s[n]$, the definition of the component i is as follows.

$$SED_s^i = \sum_{k=1}^{N} [H[k - U_i] - H[k - L_i]] g(S[k])^2, \tag{6.7}$$

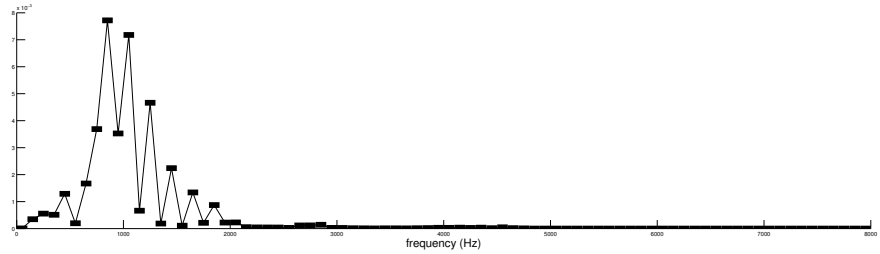
where $S[k]$ is the discrete Fourier transform of $s[n]$; $H[k]$ is the unit step function (a.k.a. the Heaviside step function); L_i and U_i indicate the lower and upper bounds of the component in the spectrum; and $g(.)$ is a normalizing function, the use of which is discussed in the remainder of this section. In this equation, N denotes the number of samples of the signal, which by principle equals the length of the signal times its sampling frequency.

Figure 6.3a and 6.3b show an arbitrary speech signal and the SED components of the signal, respectively. In Figure 6.3b, the normalizing function $g(.)$ is assumed to be the identity function, therefore the SED components form a binned power spectrum of the speech signal.

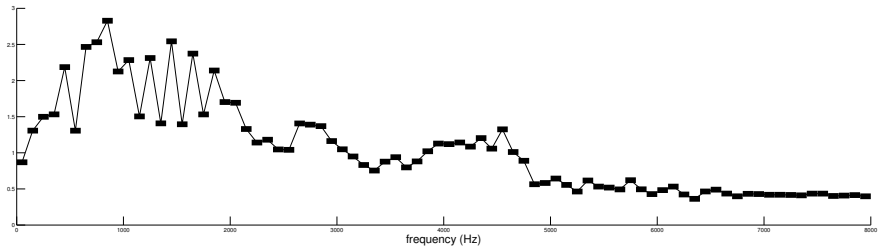
Regardless of how informative each of the components are, they can take arbitrarily large or small values. In other words, although some intervals appear to carry a relatively minor part of the energy of the signal, they can play as important of a role as others, if not



(a)



(b)



(c)

Figure 6.3: (a) A speech signal (b) SED component for $q = 1$, and (c) SED component for $q = 0.25$

more so. Therefore, as a natural solution, it is inclined to normalize the Fourier transform of the signal over the spectrum by incorporating the function $g(.)$ in the definition of the SED:

$$g(S[k]) = S^q[k];\ 0 < q \leq 1. \tag{6.8}$$

The reason why q is set to take values from $[0, 1]$ is due to the characteristic of speech

signals that their amplitude is necessarily less than or equal to one; since this property will be preserved by the discrete Fourier transform, raising to the power of q inflates $S[k]$. Figure 6.3c shows the effect of this normalization on SED components.

As for the parameter setting of SED, except the maximum value of the higher bound on the spectrum, which is dictated by the sampling frequency (Nyquist theorem), the length of each interval and the power q have to be set according to the modeling criteria.

For the purpose of our experiments, extraction of SED components is done from 100 ms windows of speech signal. The spectral interval length is set to 100 Hz. They cover from 0 to 8 kHz. The value of q is selected as 0.2. These parameters are all chosen based on a line search. The min, max, median, mean, and standard deviation of the features are used as the statistics computed over the windows of the speech signal. the dimensionality of this SED feature set is 400.

AVEC 2012 audio baseline features

The baseline features provided by AVEC 2012 [168] have the dimensionality of 1841, consisting of 25 energy- and spectral-related LLD $\times$ 42 functionals, 6 voice-related LLD $\times$ 32 functionals, 25 delta coefficients of the voice-related LLD $\times$ 19 functionals, and 10 voiced/unvoiced durational features. The details of the features and functionals are provided in Tables 4 and 5 of [168]

6.4.3 Implementation Details

Two feature sets described above have been used, i.e., SED and baseline features, as the two views v and w for a speech emotion recognition (SER) system based on the multiview SDL proposed earlier in this chapter. Hence, the two views are $\mathbf{X}^{(v)} \in \mathbb{R}^{400\times n}$ and $\mathbf{X}^{(w)} \in \mathbb{R}^{1841\times n}$, where n is 10806 in the training set and 9312 in the test set for the FCSC part of the dataset used in our experiments.

There are four dimensional affects, i.e., arousal (A), expectation (E), power (P), and Valance (V), as the *continuous* response variables to be predicted. Hence, a regressor is

to be deployed in our SER system as opposed to the FER system in Section 6.3, where a classifier was used to predict the *discrete* categorical states. The *lasso* regressor and its GLMNET[3] implementation are used in our experiments. The sparsity parameter of the *lasso* has been optimized over the training set by a 10-fold cross-validation.

An RBF kernel is used over the response variable in each dimension, which serves as the kernel over the target values ($\mathbf{L}$) to compute $\mathbf{\Phi}$ in Algorithms 3 and 4. The kernel width of the RBF kernel has been set by using a self-tuning approach similar to what is explained in [77].

In all experiments, the data in each view is normalized such that each feature is mapped to the range of [0,1]. As suggested in [168], the performance of the SER system is evaluated using Pearson's correlation coefficient (r), which is defined as follows

$$r = \frac{\sum_{i=1}^{N}(Y_i - \mu_{\mathbf{Y}})(\hat{Y}_i - \mu_{\hat{\mathbf{Y}}})}{\sqrt{\sum_{i=1}^{N}(Y_i - \mu_{\mathbf{Y}})^2}\sqrt{\sum_{i=1}^{N}(\hat{Y}_i - \mu_{\hat{\mathbf{Y}}})^2}} \tag{6.9}$$

where $\mathbf{Y}$ is the actual dimensional affect and $\hat{\mathbf{Y}}$ is the predicted one; $\mu_{\mathbf{Y}}$ and $\mu_{\hat{\mathbf{Y}}}$ are the mean of the actual and predicted dimensional affects, respectively.

6.4.4 Results

The correlation coefficient for single-view and multiview SER systems computed over the two feature sets, i.e., SED and baseline features, are reported in Table 6.5 for the arousal, expectation, power, and valence dimensions. The computation time, including the time required to learn the dictionary and coefficients, the tuning time for the sparsity coefficient of the regressor, and also the time for training and testing the regressor, is provided in Table 6.6.

As can be seen from Table 6.5, both MV1 and MV2 benefit from the complementary information in two-view features sets. The performance of the single-view system based on the SED is usually inferior to the one based on the baseline feature set. However,

[3]http://www-stat.stanford.edu/~tibs/glmnet-matlab/.

combining these two representations using one of the multiview approaches discussed earlier leads to higher correlation coefficients in all dimensions. Unlike the FER system, where MV2 clearly outperformed the MV1 approach, here these two multiview approaches have more or less similar performance. By taking the maximum correlation coefficient in each dimension and then the average over all dimensions, MV1 and MV2 achieve an average correlation coefficient of 14.73% and 14.58%, respectively, which is very close to each other. However, comparing the computation time, MV2 is faster than the MV1 approach. Unlike the facial expression recognition system, where the two views had significant differences in the dimensionality of the feature sets, here, the dimensionality of the two views is not significantly different. Hence, one feature set does not dominate the other when they are fused in the original space. Hence, MV1 can capture the complementary information of both views and performs more or less the same as MV2.

Unlike the FER system, here, most of the computation time is spent on learning the dictionary and the coefficients, due to the high number of data samples in the training set (10806). Also, here, tuning the regressor is very fast, and hence its training takes a small portion of the overall learning time of the system. Here, the computation load of the two different multiview approaches is not significantly different, as was observed in the FER system.

As a final remark, it is worth to mention that MV2 learns the dictionary and coefficients in the two views independently, and only fuses the features in the space of dictionaries learned at the final stage. It was shown that this is beneficial in the case of the FER system, where the dimensionality of one feature set is much larger than the other one and hence, fusing in original space diminishes the effect of one view. This is expected to be useful when the two views are independent or not very much correlated. If this is not the case, learning the dictionary in a fused space of two views might be beneficial, as the dictionary learned can share the common properties of both views. This can be especially useful in small dictionary size.

Table 6.5: The percentage of *correlation coefficient* (r) of the speech expression recognition system based on single-view (SV) and multiview (MV) supervised dictionary learning approaches discussed in this thesis. MV1 and MV2 are the multiview SDL approaches based on Method 1 and Method 2 as discussed in Section 6.2. Highlighted entries show the best performance in each dimension.

Dimension	Approach	Dictionary Size			
		8	16	32	64
Arousal	SV-SED	10.07	10.51	10.02	9.37
	SV-baseline	13.22	13.99	15.08	14.90
	MV1	14.30	15.39	15.70	**17.04**
	MV2	**14.61**	**15.40**	**16.66**	16.52
Expectation	SV-SED	5.19	4.61	3.66	4.12
	SV-baseline	19.56	19.30	20.31	20.73
	MV1	17.08	18.25	17.80	20.00
	MV2	**20.57**	**20.57**	**20.57**	**20.88**
Power	SV-SED	1.37	1.51	1.83	1.51
	SV-baseline	8.22	7.18	7.41	6.19
	MV1	**9.87**	7.89	**8.60**	**8.34**
	MV2	8.56	**8.21**	8.37	8.21
Valence	SV-SED	8.89	7.99	7.45	8.09
	SV-baseline	9.50	11.23	11.56	10.35
	MV1	9.82	**11.94**	12.10	**12.00**
	MV2	**11.57**	11.65	**12.20**	11.82

6.5 Summary

In this chapter, the proposed SDL approach in Chapter 3 was extended to multiview representations. Two different multiview methods were proposed: fusing the feature sets in the original space, and learning one dictionary and corresponding coefficients in this fused space (MV1), or learning one dictionary and the corresponding coefficients in each

Table 6.6: The *computation time* (in seconds) for the single-view and multiview SER systems including the time required for learning the dictionary and the coefficients, tuning the sparsity parameter for the *lasso* regressor, and eventually training and testing the regressor using tuned parameters. Highlighted entries show the fastest execution time in each dimension.

Dimension	Approach	Dictionary Size			
		8	16	32	64
Arousal	SV-SED	**33.89**	**39.27**	**90.32**	284.54
	SV-baseline	81.43	95.64	149.16	353.56
	MV1	104.61	124.56	200.89	386.51
	MV2	79.16	93.00	128.17	**134.43**
Expectation	SV-SED	**34.51**	**41.69**	**64.87**	**170.42**
	SV-baseline	82.33	97.67	141.57	295.07
	MV1	104.21	123.97	188.85	425.94
	MV2	77.53	76.80	77.51	260.16
Power	SV-SED	**36.48**	**46.69**	**70.93**	148.61
	SV-baseline	82.54	97.53	334.62	850.89
	MV1	104.27	125.58	187.21	364.10
	MV2	81.70	83.95	112.93	**87.04**
Valence	SV-SED	**33.72**	**39.55**	**64.76**	192.58
	SV-baseline	82.99	97.99	195.77	459.60
	MV1	104.64	125.19	191.97	357.63
	MV2	79.40	87.53	112.50	**107.69**

view, and then fusing the representations in the learned dictionary spaces (MV2).

It was shown that both methods benefit from the complementary information in multiple views. However, MV2 learns in the space of each view independently from others, whereas MV1 learns in the space of all views simultaneously.

The relative performance of the two proposed multiview SDL approaches was demonstrated in two emotion recognition applications, i.e., facial expression recognition (FER)

and speech emotion recognition (SER). In the former, the dimensionality of two views was such that one view dominated the other one. In this case, it was shown that MV2 performs better than MV1, as it can perform learning in the spaces of two views independently and consequently, give equal chance to both views to contribute towards the overall performance of the classification system. In SER, on the other hand, both views had the same order of dimensionality, and both MV1 and MV2 performed similarly and could capture the complementary information in both views to improve the performance over single views.

In terms of computational cost, MV2 runs faster in both applications, especially in FER. It also provides one additional parameter to tune, which is the relative dictionary sizes in multiple views. This additional parameter gives higher flexibility to this approach as it can be tuned over the training set to achieve higher performance. To avoid spending too much time on tuning this parameter, the relative size of the dictionaries in multiple views can be selected based on the relative performance of their corresponding single views, and assigning more dictionary atoms to those views with higher performance in the single view.

There are other multiview approaches in the literature, which are specifically applied to emotion recognition, such as canonical correlation analysis (CCA) [170] and cross-modal factor analysis (CFA) [121, 171]. However, the proposed multiview SDL is different from these approaches as it provides a supervised framework of multiview learning by maximizing the dependency of each view with the class labels, whereas CCA and CFA are unsupervised and perform learning by merely maximizing the correlation between two views of the data. Hence, these approaches are not directly comparable.

Chapter 7

Conclusions and Future Work

In this chapter, the summary and conclusion for the thesis is provided followed by few suggestions for the directions of future research.

7.1 Summary and Conclusion

Supervised Dictionary Learning. In this thesis a novel supervised dictionary learning was proposed. The proposed approach learns the dictionary in a space where the dependency between the data and category information is maximized. Maximizing this dependency has been performed based on the concept of the Hilbert Schmidt independence criterion (HSIC). This introduces a data decomposition that represents the data in a space with maximum dependency with category information. It was shown that the dictionary can be learned in this space in closed form. The sparse coefficients can be learned by using the *lasso* as given in (2.3). Our experiments using real-world data with varying complexity shows that the proposed approach is very efficient in classification tasks, and outperforms other unsupervised and supervised dictionary learning approaches in the literature. Also, the proposed approach is very fast and efficient in computation. However, all these improvements are achieved at the cost of more reconstruction error, which is not essential in classification tasks as the main goal is to achieve optimal discrimination not

optimal representation. The reason for increasing reconstruction error in the proposed approach is that in the optimization problem provided in (3.11), the reconstruction error is compromized to find a more discriminative dictionary.

Kernelized Supervised Dictionary Learning. Moreover, It was shown how the proposed SDL can be kernelized. This enables the proposed SDL to benefit from data-dependent kernels. It was shown using some experiments that the proposed kernelized SDL can significantly improve the results in difficult classification tasks compared to other SDL approaches in the literature. To the best of our knowledge, this is the first SDL in the literature that can be kernelized, and thus benefit from data-dependent kernels embedded into the SDL.

Properties of the Supervised Dictionary Learning. The proposed approach learns a very compact dictionary, in the sense that it significantly outperforms other approaches when the size of the dictionary is very small. This shows that the proposed SDL can effectively encode the category information into the learning of the dictionary such that it can perform very well in classification tasks using few atoms. In the dictionary learning literature, usually the dictionary learned is overcomplete, i.e., the number of elements in the learned dictionary is larger than the dimensionality of the data/dictionary. In our proposed SDL, due to the orthonormality constraint on the dictionary atoms, as in (3.11), the dictionary cannot be overcomplete. However, there are two remarks here: first, as discussed above, our dictionary is very compact and as the experiments show, the proposed SDL performs very well at small dictionary size, which is usually below even a complete dictionary size. This is a main advantage of the proposed approach, as small dictionary size means lower computational cost. Second, the kernelized version of the proposed approach can easily learn dictionaries as large as n, the number of data samples in the training set. This is because the kernel computed on the data is of the dimensionality of n, which is usually greater than the dimensionality of the data (p). Note that for all datasets provided in this paper except the Olivetti face dataset, the number of data in the training set is larger than the dimensionality of data (refer to Table 4.1). For the face dataset, it is worth noting that a dictionary as small as 32 atoms leads to extremely good results using the

proposed SDL, and overcompleteness is not necessary here.

Another advantage of the proposed approach is that there is only one parameter to be tuned, which is the regularization parameter λ in the *lasso*. Since the dictionary is learned in closed form, it is extremely fast to tune this parameter within the classification task or by minimizing the reconstruction error. Other SDL approaches in the literature usually have several parameters to be tuned, and since learning the dictionary and coefficients have to be performed alternately and iteratively, it is very time-consuming to tune these parameters using a cross-validation on the training set.

Through experimentation, it was shown that the proposed approach is less sensitive to noisy labels compared to other SDL approaches. It was also shown that by increasing the number of atoms in the dictionary, the proposed approach becomes less sensitive to noisy labels.

As mentioned above, the proposed approach is computationally efficient and fast. This is especially the case when either the dimensionality of the data (p) or the number of data samples (n) is small. In these two cases, the computation of the eigenvectors of $\mathbf{\Phi}$ given in Algorithm 1 can be performed very efficiently. However, in case that both p and n are large, efficient and fast algorithms such as recently proposed stochastic singular value decomposition (SSVD) [172] can be deployed to compute the eigenvectors of $\mathbf{\Phi}$.

Compression-Based Similarity Measure. A compression-based similarity measure was proposed designed using MPEG-1, which is a 2D compressor. It preserves spatial locality and connectivity of neighboring pixels. Through some experiments on textures, it was shown that the proposed measure works properly on both small and large patch sizes. Also, it was demonstrated that by embedding the proposed measure into the kernelized version of the introduced SDL, it outperforms the rivals in texture classification applications.

Multiview Supervised Dictionary Learning. Eventually, an extension of the proposed SDL to multiview representations was provided. Two multiview methods were formulated based on either fusing features in the original spaces and then learning the dictionary

and the coefficients (MV1), or learning one dictionary and the corresponding coefficients per view, and then fusing representations in the spaces of learned dictionaries (MV2). The relative performance of these two multiview approaches were discussed through some examples of emotion recognitions applications.

7.2 Future Work

In this research, it was proposed to use $\mathbf{L} = \mathbf{Y}^\top\mathbf{Y} + \mathbf{I}$ as the kernel on the labels. As proposed in [173, 174], it is possible to encode the relationship among the classes into a matrix $\mathbf{M} \in \mathbb{R}^{c\times c}$, where c is the number of classes, and use $\mathbf{L} = \mathbf{Y}^\top\mathbf{M}\mathbf{Y} + \mathbf{I}$ instead to build up the kernel on the labels. This may consequently better encode the data structure into the learning of the dictionary, and also reduce the sensitivity of the proposed approach to noisy labels. As a future work, this new kernel will be implemented in the formulation provided for Algorithm 1.

Also, most of the time in this research a linear kernel was used over the class labels, except for the regression problem in the speech recognition application, where an RBF kernel was used on the response variable. However, other kind of kernels can be investigated. Also, it might be possible to give weights to class labels such that when the kernel is learned over the class labels, more weight is given to some specific classes. This can be particularly useful when the data is highly imbalanced, to improve the performance of the classification system in terms of balanced classification error/accuracy.

The usefulness of data-dependent kernels was shown in a texture classification application by designing a compression-based measure and incorporating it into the proposed kernelized SDL. However, other kind of kernels can be designed in other applications and this can be one direction for future work. Particularly, it could be very interesting if similar kernels can be designed for the purpose of emotion recognition to improve its performance.

Also, an SVM with an RBF kernel was used on the sparse coefficients learned for performing the classification task. However, model selection is still an open research problem [175]. For example, the RBF kernel may not fully utilize the sparsity of the coefficients. In future work, other kernels for the SVM will be considered such that the classifier can

benefit more from the sparse nature of data points submitted for classification, as suggested in [72].

7.3 List of Publications

Several papers were published as the result of research carried out in this thesis. In the following, the list of these publications is provided.

Book Chapter

- L. Sørensen, M.J. Gangeh, S.B. Shaker, and M. de Bruijne, Texture Classification in Pulmonary CT, In: A. El-Baz and J.S. Suri (ed.) Lung Imaging and Computer Aided Diagnosis, CRC Press, 2011.

Journal Paper

- M.J. Gangeh, A. Ghodsi, and M.S. Kamel, "Kernelized Supervised Dictionary Learning", Accepted in IEEE Trans. on Signal Processing, 2013.

Refereed Conference and Workshop Papers

- M.J. Gangeh, A. Sadeghi-Naini, M.S. Kamel, and C. Czarnota, "Assessment of Cancer Therapy Effects Using Texton-Based Characterization of Quantitative Ultrasound Parametric Images", To Appear in Proceedings of International Symposium on Biomedical Imaging: From Nano to Macro, pp. 1360-1363, Apr. 2013.
- M.J. Gangeh, A. Ghodsi, and M.S. Kamel, "Supervised Texture Classification Using a Novel Compression-Based Similarity Measure", International Conference on Computer Vision and Graphics (ICCVG), LNCS, vol. 7594, pp. 379-386, Springer, Heidelberg, 2012.
- H. Parsaei, M.J. Gangeh, D.W. Stashuk, and M.S. Kamel, "Augmenting the Decomposition of EMG Signals Using Supervised Feature Extraction Techniques", Proceedings of the 34$^{\text{th}}$ International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), pp. 2615-2618, 2012.

- M.J. Gangeh, A. Ghodsi, and M.S. Kamel, "Dictionary Learning in Texture Classification", In: M. Kamel and A. Campilho (eds.), 8th International Conference on Image Analysis and Recognition (ICIAR), LNCS, vol. 6753, pp. 335-343, Springer, Heidelberg, 2011.

- M.J. Gangeh, L. Sørensen, S.B. Shaker, M.S. Kamel, and M. de Bruijne, "Multiple Classifier Systems in Texton-Based Approach for the Classification of CT Images of Lung", In Proceedings of the Medical Computer Vision: Recognition Techniques and Applications in Medical Imaging, (MCV 2010), LNCS, vol. 6533, pp. 153-163, Springer, Heidelberg, 2010.

- M.J. Gangeh, L. Sørensen, S.B. Shaker, M.S. Kamel, M. de Bruijne, and M. Loog, "A Texton-Based Approach for the Classification of Lung Parenchyma in CT Images", In: T. Jiang, N. Navab, J.P.W. Pluim, and M.A. Viergever (eds.), 13th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), LNCS, vol. 6361, pp. 595-602, Springer, Heidelberg, 2010.

- M.J. Gangeh, A.H. Shabani, and M.S. Kamel, "Nonlinear Scale Space Theory in Texture Classification Using Multiple Classifier Systems", In: A. Campilho and M. Kamel (eds.), 7th International Conference on Image Analysis and Recognition (ICIAR), LNCS, vol. 6111, pp. 147-156, Springer, Heidelberg, 2010.

- M.J. Gangeh, M.S. Kamel, and R.P.W. Duin, "Random Subspace Method in Text Categorization", 20th International Conference on Pattern Recognition (ICPR), pp. 2049-2052, 2010.

References

[1] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. New York: Springer, 2010.

[2] S. Mallat, *A Wavelet Tour of signal Processing: The Sparse Way*, 3rd ed. Academic Press, 2009.

[3] B. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, Mar. 1996.

[4] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: John Wiley & Sons, 2001.

[5] R. Tibshirani, "Regression shrinkage and selection via the *lasso*," *Journal of the Royal Statistical Society, Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[6] D. Lee and H. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, Oct. 1999.

[7] M. Biggs, A. Ghodsi, and S. Vavasis, "Nonnegative matrix factorization via rank-one downdate," in *Proceedings of the* 25^{th} *International Conference on Machine Learning (ICML)*, 2008, pp. 64–71.

[8] C. Jutten and J. Herault, "Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, no. 1, pp. 1–10, 1991.

[9] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[10] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[11] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in 27$^{\text{th}}$ *Asilomar Conference on Signals, Systems and Computers*, 1993, pp. 40–44.

[12] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[13] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, June 2010.

[14] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.

[15] R. Rubinstein, A. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, June 2010.

[16] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2008, pp. 1033–1040.

[17] M. J. Gangeh, A. Ghodsi, and M. S. Kamel, "Dictionary learning in texture classification," in *Proceedings of the* 8$^{\text{th}}$ *international conference on Image analysis and recognition - Volume Part I*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 335–343.

[18] J. Xie, L. Zhang, J. You, and D. Zhang, "Texture classification via patch-based sparse texton learning," in 17$^{\text{th}}$ *IEEE International Conference on Image Processing (ICIP)*, 2010, pp. 2737–2740.

[19] M. Varma and A. Zisserman, "A statistical approach to material classification using image patch exemplars," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2032–2047, Nov. 2009.

[20] C. Zhong, Z. Sun, and T. Tan, "Robust 3D face recognition using learned visual codebook," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–6.

[21] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[22] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in 13$^{\text{th}}$ *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 543–550.

[23] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[24] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[25] M. J. Gangeh, L. Sørensen, S. B. Shaker, M. S. Kamel, M. de Bruijne, and M. Loog, "A texton-based approach for the classification of lung parenchyma in CT images," in *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 595–602.

[26] L. Sørensen, M. J. Gangeh, S. B. Shaker, and M. de Bruijne, "Texture classification in pulmonary CT," in *Lung Imaging and Computer Aided Diagnosis*, A. El-Baz and J. S. Sure, Eds. CRC Press, 2007, pp. 343–367.

[27] M. J. Gangeh, A. Sadeghi-Naini, M. S. Kamel, and G. Czarnota, "Assessment of cancer therapy effects using texton-based characterization of quantitative ultrasound

parametric images," in *Proceedings of the International Symposium on Biomedical Imaging: From Nano to Macro (ISBI)*, 2013, pp. 1360–1363.

[28] S. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[29] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 2790–2797.

[30] E. Elhamifar, G. Sapiro, and R. Vidal, "See all by looking at a few: Sparse modeling for finding representative objects," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1600–1607.

[31] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[32] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with hilbert-schmidt norms," in *Proceedings of the* 16^{th} *international conference on Algorithmic Learning Theory (ALT)*, 2005, pp. 63–77.

[33] A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Schölkopf, "Kernel methods for measuring independence," *Journal of Machine Learning Research*, vol. 6, pp. 2075–2129, Dec. 2005.

[34] R. Cilibrasi and P. Vitányi, "Clustering by compression," *IEEE Trans. Information Theory*, vol. 51, no. 4, pp. 1523–1545, 2005.

[35] M. J. Gangeh, A. Ghodsi, and M. S. Kamel, "Supervised texture classification using a novel compression-based similarity measure," in *Proceedings of the International Conference on Computer Vision and Graphics (ICCVG)*. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 379–386.

[36] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *Advances in Neural Information Processing Systems (NIPS)*, 2007, pp. 609–616.

[37] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[38] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal Machine Learning Research*, vol. 11, pp. 19–60, Mar. 2010.

[39] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, Feb. 2010.

[40] B. Julesz, "Textons, the elements of texture perception, and their interactions," *Nature*, vol. 290, no. 5802, pp. 91–97, 1981.

[41] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision*, vol. 43, pp. 29–44, June 2001.

[42] O. Cula and K. Dana, "3d texture recognition using bidirectional feature histograms," *International Journal of Computer Vision*, vol. 59, pp. 33–60, Aug. 2004.

[43] C. Schmid, "Weakly supervised learning of visual models and its application to content-based retrieval," *International Journal of Computer Vision*, vol. 56, pp. 7–16, 2004.

[44] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *International Journal of Computer Vision: Special Issue on Texture Analysis and Synthesis*, vol. 62, no. 1-2, pp. 61–81, 2005.

[45] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Metaface learning for sparse representation based face recognition," in 17^{th} *IEEE International Conference on Image Processing (ICIP)*, 2010, pp. 1601–1604.

[46] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3501–3508.

[47] B. Fulkerson, A. Vedaldi, and S. Soatto, "Localizing objects with smart dictionaries," in *Proceedings of the* 10^{th} *European Conference on Computer Vision (ECCV): Part I*, 2008, pp. 179–192.

[48] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in 10^{th} *IEEE International Conference on Computer Vision (ICCV)*, 2005, pp. 1800–1807.

[49] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[50] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2691–2698.

[51] D.-S. Pham and S. Venkatesh, "Joint learning and dictionary construction for pattern recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[52] L. Yang, R. Jin, R. Sukthankar, and F. Jurie, "Unifying discriminative visual codebook generation with classifier training for object category recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[53] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 791–804, Apr. 2012.

[54] S. Lazebnik and M. Raginsky, "Supervised learning of quantizer codebooks by information loss minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, pp. 1294–1309, July 2009.

[55] F. Moosmann, B. Triggs, and F. Jurie, "Fast discriminative visual codebooks using randomized clustering forests," in *Advances in Neural Information Processing Systems (NIPS)*, 2006, pp. 985–992.

[56] F. Rodriguez and G. Sapiro, "Sparse representations for image classification: learning discriminative and reconstructive non-parametric dictionaries," in *IMA Preprint Series 2213*, 2007.

[57] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[58] X. Lian, Z. Li, C. Wang, B. Lu, and L. Zhang, "Probabilistic models for supervised dictionary learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2305–2312.

[59] W. Zhang, A. Surve, X. Fern, and T. G. Dietterich, "Learning non-redundant codebooks for classifying complex objects," in *Proceedings of the* 26^{th} *Annual International Conference on Machine Learning (ICML)*, 2009.

[60] F. Perronnin, "Universal and adapted vocabularies for generic visual categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1243–1256, July 2008.

[61] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New York: John Wiley & Sons, 2006.

[62] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[63] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *Journal of Machine Learning Research*, vol. 3, pp. 1–48, 2002.

[64] L. Song, J. Bedo, K. Borgwardt, A. Gretton, and A. Smola, "Gene selection via the bahsic family of algorithms," *Bioinformatics*, vol. 23, pp. i490–i498, July 2007.

[65] H. Shen, S. Jegelka, and A. Gretton, "Fast kernel-based independent component analysis," *IEEE Transactions on Signal Processing*, vol. 57, no. 9, pp. 3498–3511, Sept. 2009.

[66] N. Quadrianto, A. Smola, L. Song, and T. Tuytelaars, "Kernelized sorting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1809–1821, Oct. 2010.

[67] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample problem," Max Planck Institute for Biological Cybernetics, Technical Report 157, Apr. 2008.

[68] S. Jegelka, A. Gretton, B. Schölkopf, B. K. Sriperumbudur, and U. Von Luxburg, "Generalized clustering via kernel embeddings," in *Proceedings of the* 32$^{\text{nd}}$ *Annual German Conference on Advances in Artificial Intelligence*, 2009, pp. 144–152.

[69] J. Marden, *Analyzing and Modeling Rank Data*. London: Chapman and Hall, 1995, page 59.

[70] H. Lütkepohl, *Handbook of Matrices*. John Wiley & Sons, 1996.

[71] J. L. Alperin, *Local Representation Theory: Modular Representations as an Introduction to the Local Representation Theory of Finite Groups*. New York: Cambridge University Press, 1986.

[72] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *Proceedings of the* 24$^{\text{th}}$ *international conference on Machine learning (ICML)*, 2007, pp. 759–766.

[73] *Cambridge University Computer Laboratory, Olivetti Face Dataset AT&T*, 1994, http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[74] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit," Dept. of Computer Science, Technion, Technical Report, 2008.

[75] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems*. Morgan Kaufmann, 1990, pp. 396–404.

[76] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–8.

[77] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in Neural Information Processing Systems (NIPS)*, 2004, pp. 1601–1608.

[78] E. Pąkalska and R. Duin, *The Dissimilarity Representation for Pattern Recognition: Foundations And Applications*. Singapore: World scientific, 2005.

[79] E. Pąkalska, P. Paclik, and R. Duin, "A generalized kernel approach to dissimilarity-based classification," *Journal of Machine Learning Research*, vol. 2, pp. 175–211, Mar. 2002.

[80] E. Pąkalska and R. Duin, "Dissimilarity representations allow for building good classifiers," *Pattern Recognition Letters*, vol. 23, pp. 943–956, 2002.

[81] M. Mirmehdi, X. Xie, and E. J. Suri, *Handbook of Texture Analysis*. London: Imperial Collage Press, 2008.

[82] M. Garcia and D. Puig, "Supervised texture classification by integration of multiple texture methods and evaluation windows," *Image and Vision Computing*, vol. 25, no. 7, pp. 1091–1106, 2007.

[83] T. Randen and J. Husøy, "Filtering for texture classification: A comparative study," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 291–310, 1999.

[84] M. Li, X. Chen, X. Li, B. Ma, and P. Vitányi, "The similarity metric," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3250–3264, 2004.

[85] A. Bardera, M. Feixas, I. Boada, and M. Sbert, "Image registration by compression," *Information Sciences*, vol. 180, no. 7, pp. 1121–1133, 2010.

[86] X. Zhang, Y. Hao, X. Zhu, and M. Li, "Information distance from a question to an answer," in *Proceedings of the* 13^{th} *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD, 2007, pp. 874–883.

[87] M. Li and P. Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications*, 3rd ed. New York: Springer, 2008.

[88] B. Campana and E. Keogh, "A compression-based distance measure for texture," *Statistical Analysis and Data Mining*, vol. 3, no. 6, pp. 381–398, 2010.

[89] J. Mortensen, J. Wu, J. Furst, J. Rogers, and D. Raicu, "Effect of image linearization on normalized compression distance," in *Signal Processing, Image Processing and Pattern Recognition*, ser. Communications in Computer and Information Science, D. Slezak, S. K. Pal, B. H. Kang, J. Gu, H. Kuroda, and T. H. Kim, Eds., 2009, vol. 61, pp. 106–116.

[90] A. Macedonas, D. Besiris, G. Economou, and S. Fotopoulos, "Dictionary based color image retrieval," *Journal of Visual Communication and Image Representation*, vol. 19, no. 7, pp. 464–470, 2008.

[91] D. Cerra and M. Datcu, "Image retrieval using compression-based techniques," in *Int. ITG Conf. Source and Channel Coding (SCC), 2010*, 2010, pp. 1–6.

[92] D. Cerra, A. Mallet, L. Gueguen, and M. Datcu, "Algorithmic information theory-based analysis of earth observation images: An assessment," *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 1, pp. 8–12, 2010.

[93] P. Vázquez and J. Marco, "Using normalized compression distance for image similarity measurement: an experimental study," *The Visual Computer*, pp. 1–22, 2011.

[94] M. Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding*. London, UK: The Institution of Electrical Engineers, 2003.

[95] M. Petrou and P. G. Sevilla, *Image Processing Dealing with Texture*. West Sussex: John Wiley & Sons, 2006.

[96] E. Hadjidemetriou, M. Grossberg, and S. Nayar, "Multiresolution histograms and their use for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 831–847, 2004.

[97] T. Ahonen and M. Pietikainen, "Image description using joint distribution of filter bank responses," *Pattern Recognition Letters*, vol. 30, no. 4, pp. 368–376, 2009.

[98] U. Kandaswamy, D. Adjeroh, and M. Lee, "Efficient texture analysis of sar imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 9, pp. 2075–2083, Sept. 2005.

[99] M. J. Gangeh, L. Sørensen, S. B. Shaker, M. S. Kamel, and M. de Bruijne, "Multiple classifier systems in texton-based approach for the classification of CT images of lung," in *Proceedings of the Medical Computer Vision: Recognition Techniques and Applications in Medical Imaging (MCV)*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 153–163.

[100] I. Sluimer, P. van Waes, M. Viergever, and B. van Ginneken, "Computer-aided diagnosis in high resolution CT of the lungs," *Medical Physics*, vol. 30, no. 12, pp. 3081–3090, 2003.

[101] I. Sluimer, M. Prokop, I. Hartmann, and B. van Ginneken, "Automated classification of hyperlucency, fibrosis, ground glass, solid, and focal lesions in high-resolution CT of the lung," *Medical Physics*, vol. 33, no. 7, pp. 2610–2620, 2006.

[102] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision*, vol. 81, pp. 2–23, 2009.

[103] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, 1998, pp. 92–100.

[104] M. R. Amini, N. Usunier, and C. Goutte, "Learning from multiple partially observed views - an application to multilingual text categorization," in *Advances in Neural Information Processing Systems (NIPS)*, 2009, pp. 28–36.

[105] Y. Wang and L. Guan, "Recognizing human emotional state from audiovisual signals," *IEEE Transactions on Multimedia*, vol. 10, no. 5, pp. 936–946, Aug. 2008.

[106] B. Schuller, M. Valstar, F. Eyben, , R. Cowie, and M. Pantic, "AVEC 2012 -the continuous audio/visual emotion challenge," in *ACII*, 2012.

[107] X. Zhao, N. Evans, and J.-L. Dugelay, "Multi-view semi-supervised discriminant analysis: A new approach to audio-visual person recognition," in *European Signal Processing Conference (EUSIPCO)*, 2012, pp. 27–31.

[108] T. Joshi, S. Dey, and D. Samanta, "Multimodal biometrics: state of the art in fusion techniques," *International Journal of Biometrics*, vol. 1, no. 4, pp. 393–417, Jul. 2009.

[109] H. D. I. Abhishek Kumar, Piyush Rai, "Co-regularized multi-view spectral clustering," in *Advances in Neural Information Processing Systems (NIPS)*, 2011, pp. 1413–1421.

[110] M. Chen, K. Q. Weinberger, and J. C. Blitzer, "Co-training for domain adaptation," in *Advances in Neural Information Processing Systems (NIPS)*, 2011, pp. 2456–2464.

[111] X. Z. D. S. Martha White, Yao-Liang Yu, "Convex multi-view subspace learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1682–1690.

[112] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, New Jersey: John Wiley & Sons, 2004.

[113] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[114] G. Brown, "Ensemble learning," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Springer, 2010.

[115] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on local binary patterns: A comprehensive study," *Image and Vision Computing*, vol. 27, no. 6, pp. 803–816, May 2009.

[116] L. Stan and A. K. Jain, *Handbook of Face Recognition*, 2nd ed. Springer, 2005.

[117] B. Fasel and J. Luettin, "Automatic facial expression analysis: a survey," *Pattern Recognition*, vol. 36, no. 1, pp. 259–275, 2003.

[118] P. Ekman, *Basic Emotions*. Sussex, U.K.: John Wiley & Sons, Ltd, 1999.

[119] Y. Tie and L. Guan, "A deformable 3D facial expression model for dynamic human emotional state recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 1, pp. 142–157, Jan. 2013.

[120] ——, "Human emotional state recognition using real 3D visual features from Gabor library," *Pattern Recognition*, vol. 46, no. 2, pp. 529–538, 2013.

[121] Y. Wang, L. Guan, and A. Venetsanopoulos, "Kernel cross-modal factor analysis for information fusion with application to bimodal emotion recognition," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 597–607, Jun. 2012.

[122] S. Liu, Q. Ruan, C. Wang, and G. An, "Tensor rank one differential graph preserving analysis for facial expression recognition," *Image and Vision Computing*, vol. 30, no. 8, pp. 535–545, 2012.

[123] R. Araujo, Y.-Q. Miao, M. S. Kamel, and M. Cheriet, "A fast and robust feature set for cross individual facial expression recognition," in *Proceedings of the International Conference on Computer Vision and Graphics (ICCVG)*. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 272–279.

[124] M. Pantic and L. Rothkrantz, "Facial action recognition for facial expression analysis from static face images," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 34, no. 3, pp. 1449–1461, Jun. 2004.

[125] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[126] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.

[127] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2010, pp. 94–101.

[128] T. Kanade, J. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in 4th *IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 46–53.

[129] M. Lyons, J. Budynek, and S. Akamatsu, "Automatic classification of single facial images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 12, pp. 1357–1362, Dec. 1999.

[130] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in *Proceedings of the* 3rd *International Conference on Face & Gesture Recognition*, 1998, pp. 200–205.

[131] M. Grimm, K. Kroschel, and S. Narayanan, "The Vera am Mittag German audio-visual emotional speech database," in *IEEE International Conference on Multimedia and Expo (ICME)*, Apr. 2008, pp. 865–868.

[132] P. Viola and M. Jones, "Robust real-time object detection," in 2nd *International Workshop on Statistical and Computational Theories of Vision*, 2001.

[133] C. Caffi and R. W. Janney, "Toward a pragmatics of emotive communication," *Journal of Pragmatics*, vol. 22, no. 3 - 4, pp. 325–373, 1994.

[134] R. W. Picard, "Affective computing for HCI," *Human-Computer Interaction: Ergonomics and User Interfaces*, vol. 1, pp. 829–833, 1999.

[135] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. Taylor, "Emotion recognition in human-computer interaction," *Signal Processing Magazine, IEEE*, vol. 18, no. 1, pp. 32–80, Jan. 2001.

[136] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, pp. 1161–1178, 1980.

[137] A. Mehrabian, "Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament," *Current Psychology*, vol. 14, pp. 261–292, 1996.

[138] J. Fontaine, K. Scherber, E. Roesch, and P. Ellsworth, "The world of emotions is not two-dimensional," *Psychological Science*, vol. 18, no. 12, pp. 1050–1057, 2007.

[139] B. Schuller, M. Valstar, F. Eyben, G. McKeown, R. Cowie, and M. Pantic, "AVEC 2011–the first international audio/visual emotion challenge," in *Affective Computing and Intelligent Interaction*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2011, vol. 6975, pp. 415–424.

[140] B. Schuller, S. Steidl, and A. Batliner, "The interspeech 2009 emotion challenge," in *Interspeech*, 2009, pp. 312–315.

[141] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Müller, and S. S. Narayanan, "The interspeech 2010 paralinguistic challenge," in *Interspeech*, 2010, pp. 2794–2797.

[142] R. Calix, M. Khazaeli, L. Javadpour, and G. Knapp, "Dimensionality reduction and classification analysis on the audio section of the semaine database," in *Affective Computing and Intelligent Interaction*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2011, vol. 6975, pp. 323–331.

[143] L. Cen, Z. Yu, and M. Dong, "Speech emotion recognition system based on L1 regularized linear regression and decision fusion," in *Affective Computing and Intelligent*

Interaction, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2011, vol. 6975, pp. 332–340.

[144] S. Pan, J. Tao, and Y. Li, "The CASIA audio emotion recognition method for audio/visual emotion challenge 2011," in *Affective Computing and Intelligent Interaction*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2011, vol. 6975, pp. 388–395.

[145] G. Ramirez, T. Baltrušaitis, and L.-P. Morency, "Modeling latent discriminative dynamic of multi-dimensional affective signals," in *Affective Computing and Intelligent Interaction*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2011, vol. 6975, pp. 396–406.

[146] A. Sayedelahl, P. Fewzee, M. Kamel, and F. Karray, "Audio-based emotion recognition from natural conversations based on co-occurrence matrix and frequency domain energy distribution features," in *Affective Computing and Intelligent Interaction*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2011, vol. 6975, pp. 407–414.

[147] R. Sun and E. Moore, "Investigating glottal parameters and teager energy operators in emotion recognition," in *Affective Computing and Intelligent Interaction*, ser. Lecture Notes in Computer Science, S. D'Mello, A. Graesser, B. Schuller, and J.-C. Martin, Eds. Springer Berlin / Heidelberg, 2011, vol. 6975, pp. 425–434.

[148] J. Kim, H. Rao, and M. Clements, "Investigating the use of formant based features for detection of affective dimensions in speech," in *Affective Computing and Intelligent Interaction*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2011, vol. 6975, pp. 369–377.

[149] H. Meng and N. Bianchi-Berthouze, "Naturalistic affective expression classification by a multi-stage approach based on hidden markov models," in *Affective Computing and Intelligent Interaction*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2011, vol. 6975, pp. 378–387.

[150] P. Fewzee and F. Karray, "Elastic net for paralinguistic speech recognition," in *Proceedings of the* 14th *ACM International Conference on Multimodal Interaction*, 2012, pp. 509–516.

[151] Y.-H. Yang, Y.-C. Lin, Y.-F. Su, and H. Chen, "Music emotion classification: A regression approach," in *IEEE International Conference on Multimedia and Expo*, July 2007, pp. 208–211.

[152] H. Espinosa, C. Garcia, and L. Pineda, "Bilingual acoustic feature selection for emotion estimation using a 3d continuous model," in *IEEE International Conference on Automatic Face Gesture Recognition and Workshops*, Mar. 2011, pp. 786–791.

[153] A. Tarasov and S. Delany, "Benchmarking classification models for emotion recognition in natural speech: A multi-corporal study," in *IEEE International Conference on Automatic Face Gesture Recognition and Workshops*, Mar. 2011, pp. 841–846.

[154] J. Rong, G. Li, and Y.-P. P. Chen, "Acoustic feature selection for automatic emotion recognition from speech," *Information Processing and Management*, vol. 45, no. 3, pp. 315–328, 2009.

[155] D. Morrison, R. Wang, and L. C. D. Silva, "Ensemble methods for spoken emotion recognition in call-centres," *Speech Communication*, vol. 49, no. 2, pp. 98–112, 2007.

[156] O. Pierre-Yves, "The production and recognition of emotions in speech: features and algorithms," *International Journal of Human-Computer Studies*, vol. 59, no. 1-2, pp. 157–183, 2003.

[157] E. Fersini, E. Messina, and F. Archetti, "Emotional states in judicial courtrooms: An experimental investigation," *Speech Communication*, vol. 54, no. 1, pp. 11–22, 2012.

[158] B. Schuller, G. Rigoll, and M. Lang, "Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, May 2004, pp. 577–580.

[159] D. Neiberg, K. Elenius, and K. Laskowski, "Emotion recognition in spontaneous speech using gmms," in *Interspeech*, 2006.

[160] C. Yu, P. M. Aoki, and A. Woodruff, "Detecting user engagement in everyday conversations," in *International Conference on Spoken Language Processing (ICSLP)*, vol. 2, Oct. 2004, pp. 1329–1332.

[161] E. Douglas-Cowie, R. Cowie, C. Cox, N. Amier, and D. K. J. Heylen, "The sensitive artificial listner: an induction technique for generating emotionally coloured conversation," in *LREC Workshop on Corpora for Research on Emotion and Affect*, 2008, pp. 1–4.

[162] G. McKeown, M. Valstar, R. Cowie, and M. Pantic, "The SEMAINE corpus of emotionally coloured character interactions," in *IEEE International Conference on Multimedia and Expo (ICME)*, July 2010, pp. 1079–1084.

[163] G. McKeown, M. Valstar, R. Cowie, M. Pantic, and M. Schroder, "The semaine database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent," *IEEE Transactions on Affective Computing*, vol. 3, no. 1, pp. 5–17, 2012.

[164] A. Cruz, B. Bhanu, and S. Yang, "A psychologically-inspired match-score fusion model for video-based facial expression recognition," in *Affective Computing and Intelligent Interaction*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2011, vol. 6975, pp. 341–350.

[165] M. Dahmane and J. Meunier, "Continuous emotion recognition using gabor energy filters," in *Affective Computing and Intelligent Interaction*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2011, vol. 6975, pp. 351–358.

[166] M. Glodek, S. Tschechne, G. Layher, M. Schels, T. Brosch, S. Scherer, M. Kächele, M. Schmidt, H. Neumann, G. Palm, and F. Schwenker, "Multiple classifier systems for the classification of audio-visual emotional states," in *Affective Computing and Intelligent Interaction*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2011, vol. 6975, pp. 359–368.

[167] F. Eyben, M. Wollmer, M. Valstar, H. Gunes, B. Schuller, and M. Pantic, "String-based audiovisual fusion of behavioural events for the assessment of dimensional affect," in *IEEE International Conference on Automatic Face Gesture Recognition and Workshops*, Mar. 2011, pp. 322–329.

[168] B. Schuller, M. Valstar, F. Eyben, R. Cowie, and M. Pantic, "AVEC 2012 – the continuous audio/visual emotion challenge," in *Proceedings of the* 14$^{\text{th}}$ *ACM international conference on Multimodal interaction*. ACM, 2012, pp. 449–456.

[169] P. Fewzee and F. Karray, "Emotional speech: A spectral analysis," in *Proceedings of Interspeech*, Sept. 2012.

[170] H. Hoteling, "Relations between two sets of variates," *Biometrika*, vol. 28, pp. 312–377, 1936.

[171] D. Li, N. Dimitrova, M. Li, and I. K. Sethi, "Multimedia content processing through cross-modal association," in *Proceedings of the* 11$^{\text{th}}$ *ACM international conference on Multimedia*, 2003, pp. 604–611.

[172] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, May 2011.

[173] M. Blaschko and A. Gretton, "Taxonomy inference using kernel dependence measures," Max Planck Institute for Biological Cybernetics, Technical Report 181, Nov. 2008.

[174] L. Song, A. Smola, A. Gretton, and K. Borgwardt, "A dependence maximization view of clustering," in *Proceedings of the* 24$^{\text{th}}$ *international conference on Machine learning (ICML)*, 2007, pp. 815–822.

[175] D. Anguita, A. Ghio, L. Oneto, and S. Ridella, "In-sample and out-of-sample model selection and error estimation for support vector machines," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 9, pp. 1390–1406, Sept. 2012.