SPARSE REPRESENTATION FOR FACE RECOGNITION

A DISSERTATION

Submitted in partial fulfillment of
the requirements for the award of the degree
of

INTEGRATED DUAL DEGREE
in
ELECTRONICS AND COMMUNICATION ENGINEERING
(With Specialization in Wireless Communication)

By
NEERAJ GANGWAR

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE (INDIA) - 247 667

JUNE 2014

ACKNOWLEDGMENT

Foremost, I would like to express my gratitude to my guide Dr. Debashis Ghosh for his constant support, inspiration and guidance. I am fortunate to have been his student and thank him for his constant motivation and valuable advice. Next, I would like to thank my teachers especially Dr. D. K. Mehra, Mr. S. Chakravorty, Dr. S. N. Sinha and Dr. M. V. Kartikeyan for their valuable lectures. I wish to express my gratitude to Dr. Vinod Pankajakshan for his valuable perspective and feedback.

I have been fortunate to work with a talented group of people of Signal Processing Laboratory, IIT Roorkee and would like to thank everyone for all their help and valuable suggestions. I could have not asked for a more interesting group to work with.

I would like to thank all my friends especially Karan, Puneet, Giri, Siva, Bhawna, Swati, Shivangi, Deshank, Rahul, Amit and Madhur for making my stay at IIT Roorkee memorable.

Finally, I wish to express my heartfelt gratitude to a special group of people: my parents, Shri Virendra Gangwar and Smt. Rajeshwari, and my brother, Ashu. Any success of mine would be impossible without your love and encouragement.

Abstract

Sparse representation has attracted a great deal of attention in the past decade. Famous transforms such as discrete Fourier transform, wavelet transform and singular value decomposition are used to sparsely represent the signals. The aim of these transforms is to reveal certain structures of a signal and representation of these structures in a compact form. Therefore, sparse representation provides high performance in the areas as diverse as image denoising, pattern classification, compression etc. All of these applications are concerned with a compact and high-fidelity representation of signals.

In this thesis, we consider the classical face recognition problem. This application is more concerned with the semantic information of image signals. It is shown that a sparse representation based framework is a possible way to tackle this problem. We also propose a new approach for face classification which is based on task driven dictionary learning.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BP	Basis Pursuit
CS	Compressed Sensing
GA	Greedy Algorithms
$LARS$	Least Angle Regression
$LASSO$	Least Absolute Shrinkage and Selection Operator
$LCKSVD$	Label Consistent K-SVD
LP	Linear Programming
MAP	Maximum A-posteriori Probability
ML	Maximum Likelihood
MCA	Morphological Component Analysis
MP	Matching Pursuit
NN	Nearest Neighbor
NS	Nearest Subspace
OMP	Orthogonal Matching Pursuit
PCA	Principal Component Analysis
SRC	Sparse Representation based Classification
SVD	Singular Value Decomposition
SVM	Support Vector Machine

CHAPTER 1

INTRODUCTION

The main accomplishment of classical linear algebra is a thorough examination of the systems of linear equations. Linear system of equations has applications in many engineering developments and solutions. Recently, sparse solutions of linear systems have attracted plenty of attention from not only mathematicians but also engineers as this field has many amazing applications in image and signal processing.

Formally, a system of linear equations can be represented by a matrix equation

$$\mathbf{y} = \mathbf{D}\mathbf{x} \tag{1.1}$$

where $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{D} \in \mathbb{R}^{n \times k}$ and $\mathbf{x} \in \mathbb{R}^k$. n and k are generally referred to as the number of observations and the number of degrees of freedom respectively [1].

- If $n = k$, the number of equations is equal to the number of variables. In this case, assuming $\mathbf{D}$ is a full rank matrix[1], this system has a unique solution.

$$\mathbf{x} = \mathbf{D}^{-1}\mathbf{y} \tag{1.2}$$

- If $n > k$, the number of equations exceeds the number of variables. In this case, the columns don't span the entire $\mathbb{R}^n$ space. This system has a unique solution if $\mathbf{y}$ resides in the column space of $\mathbf{D}$. If this is not the case, this system has no solution and it is desirable to solve

$$\arg\min_{\mathbf{x}} \quad \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \tag{1.3}$$

[1] A square matrix is said to have full rank if its columns span the entire $\mathbb{R}^n$ space. In this case, the matrix is non-singular and its inverse exists.

which leads to an approximate solution

$$\hat{\mathbf{x}} = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{y} = \mathbf{D}^+\mathbf{y} \tag{1.4}$$

- If $n < k$, the number of equations becomes less than the number of variables. Assuming $\mathbf{D}$ to be a full row rank[2] matrix, this system has infinitely many solutions.

In engineering, often problems involving underdetermined systems are encountered. An example is the image scale up problem where due to blur and scale down operations, an unknown image $\mathbf{x}$ results in a lower quality image $\mathbf{y}$. The matrix $\mathbf{D}$ represents degradation operations. The aim is to recover the original image $\mathbf{x}$ from $\mathbf{y}$.

1.1 Underdetermined Systems

In all the problems involving underdetermined systems, a single solution is desired. Since an underdetermined system of equations has infinitely many solutions, additional constraints have to be imposed to find a unique solution. A simple way to achieve this goal is *regularization*. This optimization problem involves an objective function $J(\mathbf{x})$ that evaluates the desirability of a would-be solution and is defined as

$$\arg\min_{\mathbf{x}} \quad J(\mathbf{x}) \quad \text{subject to} \quad \mathbf{y} = \mathbf{D}\mathbf{x} \tag{1.5}$$

For the image scale up problem, typically a function $J(\mathbf{x})$ that prefers smooth or piecewise smooth solutions is used.

A very famous choice for $J(\mathbf{x})$ is the Euclidean norm of the solution. In this case, $J(\mathbf{x}) = \|\mathbf{x}\|_2$. This problem leads to the solution

$$\hat{\mathbf{x}} = \mathbf{D}^T\left(\mathbf{D}\mathbf{D}^T\right)^{-1}\mathbf{y} \tag{1.6}$$

The reason for the popularity of the Euclidean norm is that it leads to a unique solution. It is not just the Euclidean norm that gives a unique solution, any convex function $J(\cdot)$ promises this uniqueness. This includes all the l_p norms[3] for $p \geq 1$

$$\|\mathbf{x}\|_p^p = \sum_{i=1}^{k} |x_i|^p \tag{1.7}$$

An important thing to note is that the l_p norm with $1 \leq p \leq \infty$ without the p^{th} power is a non-strict convex function. In this case, uniqueness is not guaranteed.

[2] A rectangular matrix is said to have full row rank if its columns span the entire $\mathbb{R}^n$ space.

[3] l_p norms are formally defined in the next chapter.

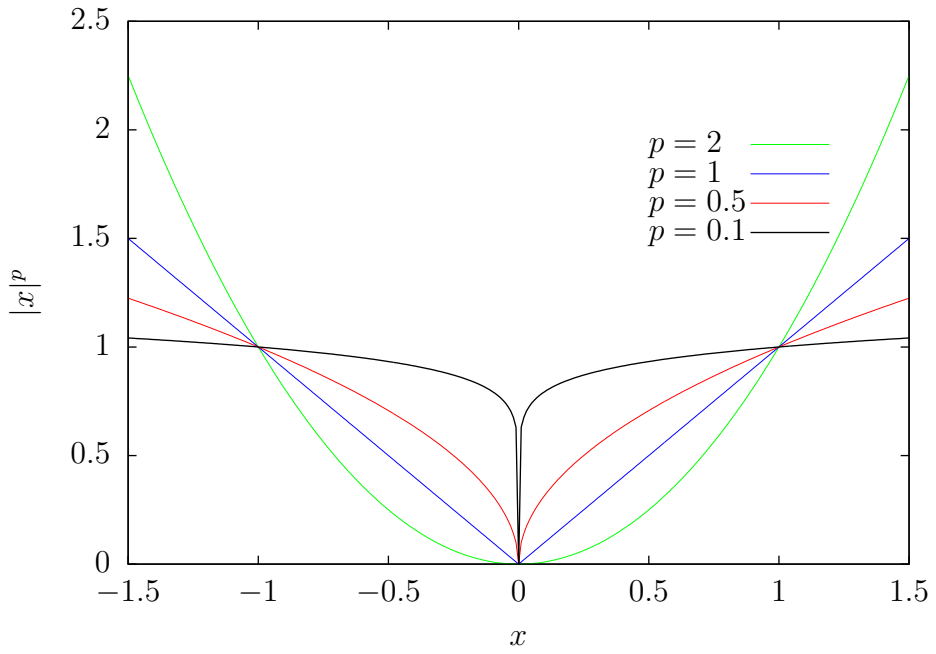


Figure 1.1: $|x|^p$ for various values of p

The l_∞ norm and the l_1 norm are very popular. In this thesis, the l_1 norm will be of interest as it tends to sparsify the solution. The l_1 norm is not strictly convex, so it may have more than one solution. Nevertheless, it has been proved that there would exist atleast one solution with atmost n non-zero entries [2]. It means that as one moves from l_2 towards l_1 norm, one promotes sparser solutions. This compelled researchers to consider l_p norms with $p < 1$.

1.2 Sparse Representation

The extreme case for $p < 1$ is the case when $p \to 0$. For $J(\mathbf{x}) = \|\mathbf{x}\|_0$, equation (1.5) reduces to the sparse representation problem and can be described as

$$P_0 : \quad \arg\min_{\mathbf{x}} \quad \|\mathbf{x}\|_0 \quad \text{subject to} \quad \mathbf{y} = \mathbf{D}\mathbf{x} \tag{1.8}$$

Figure 1.1 shows the plot of $|x|^p$ for various values of p. It can be seen that as $p \to 0$, $|x|^p$ approaches the indicator function i.e. it takes a value of 0 if x is zero and 1 otherwise. This means that the l_0 norm measures the number of non-zero entries in $\mathbf{x}$. Problem defined in (1.8) seeks for a solution with the minimum number of non-zero entries. While the l_0 norm gives a simple notion of sparsity, the standard convex analysis is not applicable.

At this point, two questions become very important

1. Does a unique solution of P_0 exist?
2. If it does, under what conditions?

These questions are answered in the next chapter where uniqueness of solution is covered in detail.

1.3 Sparse Decomposition Algorithms

Sparse decomposition algorithms are designed to solve (1.8) and find a vector $\mathbf{x}$ with the least number of non-zero entries. It can be noted that $\mathbf{x}$ has two important parts - the support of the solution and the entries over this support. Over the years, various algorithms have been proposed to compute the sparse representation of a given signal. Some of them are as follows

1. **Convex Relaxation Techniques:** The problem defined by equation (1.8) is non-convex in nature and NP hard. One way to tackle this problem is to replace the l_0 norm with a smooth penalty function. Basis pursuit replaces l_0 norm with l_1 norm to make the problem convex. It is also known as l_1 minimization method. Under certain conditions, both the problems converge to the same solution.

   Donoho and Huo showed empirically in [3] that for a dictionary $\mathbf{D}$ that is made by concatenation of Fourier and spikes bases, the solution of P_0 can be found using basis pursuit. Later, Donoho and Elad in [4] generalized these results for any dictionary $\mathbf{D}$ that can be formed by concatenating several bases or even less structured systems. These results are presented in detail in chapter 3.

2. **Greedy Algorithms:** Greedy algorithms (GA) give an approximate solution of (1.8). These algorithms focus on the support of the solution. Once the support is found, the entries of the solution can be computed by solving least squares. In [5], it was shown that matching pursuit (MP) computes a compact representation quickly. These methods have simple and fast implementations when compared with basis pursuit. It was also shown that the error initially decreases rapidly but asymptotic decay rate becomes much slower for GA.

   A variation of matching pursuit, orthogonal matching pursuit (OMP) was first proposed for wavelet decomposition. This method gives significantly better performance than MP [6]. It was also shown that OMP recovers a solution if the l_0 norm of the solution is less than $\frac{1}{2}\left(\mu^{-1}+1\right)$, here μ is the mutual coherence of $\mathbf{D}$. OMP also reliably recovers a k dimensional solution with k_0 non-zero entries from $O(k_0 \ln k)$ random linear measurements [7].

   Though these methods give simple and fast implementations, they are sub-optimal in nature and sometimes fail to give the correct solution. Other famous methods based on MP are regularized OMP (ROMP), stepwise OMP (StOMP) etc [8].

In this thesis, algorithms from only these two families are used for the computation of the sparse coefficients for a given signal. The choice of algorithm depends on the requirements.

1.4 Dictionary Learning

Sparse decomposition algorithms assume that the dictionary $\mathbf{D}$ is given as an input. The choice of dictionary plays a very crucial role in finding a good sparse representation of a signal $\mathbf{y}$. For years, orthogonal and bi-orthogonal dictionaries attracted a lot of attention due to their mathematical simplicity. Examples of these dictionaries are Fourier, wavelets etc. Fourier representation is suitable for projecting a signal onto the k_0 lowest frequency atoms. It becomes difficult to represent a discontinuity using Fourier basis as the representation contains large coefficients over all frequencies. Moreover, these dictionaries are generally suitable only for 1-d signals and inadequate in representing more complex signals.

To satisfy the rising needs, a variety of dictionaries were developed which come from one of the two sources [9]

1. Mathematical model of data which leads to analytic dictionaries.
2. A set of realization of data which leads to more flexible dictionaries that can adapt to specific signal data. These dictionaries are termed as learned dictionaries.

The first category of dictionaries covers wavelets, steerable wavelets, curvelets and more. These dictionaries usually feature fast implicit implementations. The main limitation of these dictionaries is that they are only as successful as the underlying model of the data.

With the recent development in the field of machine learning, it has been observed that the structure of complex natural phenomenon can be extracted more accurately from the data itself. This belief led to the development of the learned dictionaries. Initial work in this field started with the probabilistic methods for dictionary learning [10]. These methods include maximum likelihood (ML) dictionary learning and maximum a-posteriori (MAP) dictionary learning.

Both ML and MAP based dictionary learning algorithms consider an overcomplete dictionary as a probabilistic model of the data. ML based algorithm [11] maximizes the likelihood probability of the data $P(\mathbf{X} \mid \mathbf{D})$ whereas MAP based algorithm [12] maximizes the a-posteriori probability of the dictionary $P(\mathbf{D} \mid \mathbf{X})$. These algorithms solve the problem iteratively in two steps - sparse approximation and the learning step. This two step optimization structure has been preserved in most of later work in the field. For example, method of optimal directions (MOD) uses the two step iterative procedure to learn the dictionary [13].

Aharon et al. proposed a new algorithm for dictionary learning known as the K-SVD algorithm. This algorithm is a generalization of the K-means algorithm [14]. This algorithm differs from earlier

ones in a way that the dictionary atoms are handled separately. It considers one dictionary atoms at a time and updates it with the new one along with the coefficients that multiply it in $\mathbf{X}$.

1.5 Applications

Initial applications of sparse representation include the ones in which equation (1.1) was interprated as a way of reconstructing the signals. The very first application of sparse model is compression. Here a signal $\mathbf{y}$ is approximated as a linear combination of k_0 out of k dictionary atoms. This signal now can be expressed using $2k_0$ scalars - k_0 scalars for the positions of non-zero entries and rest for the actual values of these non-zero entries.

Another application of sparse model is compressive sensing (CS). CS relies on the belief that the signals that are sparsely generated, can be recovered from a reduced number of measurements [15]. If a signal $\mathbf{f}$ has a sparse representation $\mathbf{x}$ over a dictionary $\mathbf{\Psi} \in \mathbb{R}^{m \times k}$ and the measurement matrix is defined as $\mathbf{\Phi} \in \mathbb{R}^{n \times m}$, then measurements $\mathbf{y}$ can be defined as

$$\mathbf{y} = \mathbf{\Phi f} = \mathbf{\Phi \Psi x} \tag{1.9}$$

If $m \ll n$, this system becomes an underdetermined system and if $\mathbf{x}$ is sparse, it can be recovered, leading to the recovery of the original signal with a high probability.

Other applications include image denoising, morphological component analysis (MCA) etc. This thesis considers the applications in the field of computer vision and pattern recognition. Specifically, we consider the classical problem of face recognition.

In this thesis, we discuss a sparse representation based classification framework that can be considered as a general classification algorithm for object recognition. It can be considered as a generalization of nearest neighbor (NN) and nearest subspace (NS) classifiers. It'll be shown that this framework outperforms these classification techniques.

1.6 Objective

From the discussion so far, it is clear that sparse representation has three main branches

- Sparse decomposition algorithms
- Dictionary learning
- Applications

The work presented in this thesis covers the following

1. Study of underlying theory of sparse representation and various sparse decomposition algorithms such as orthogonal matching pursuit, basis pursuit, LASSO, LARS etc.
2. Study of various dictionary learning algorithms such as ML dictionary learning, MAP dictionary learning, method of optimal directions and the K-SVD algorithm.
3. Study of discriminative dictionary learning algorithms such as LC-KSVD and task driven dictionary learning and their applications in the field of face recognition.

The main focus of the work presented in this thesis is on classification using sparse representation. We consider only the classification aspect of the face recognition problem throughout this thesis. We also modify the LC-KSVD algorithm by changing the classification approach used. Following this, a new approach for face recognition is presented that is based on discriminative dictionary learning and SRC.

1.7 Organization of the Thesis

In chapter 2 of this thesis, the preliminaries for sparse representation is presented. This chapter covers the notations and; useful definitions and theorems; that underlie the content of this thesis. Chapter 3 reviews various sparse decomposition algorithms that are used in this thesis. Chapter 4 focuses on a new framework for classification that is based on sparse representation. This chapter also includes benchmark results for this framework. Chapter 5 deals with the problem of dictionary learning and focuses on various dictionary learning algorithms. Chapter 6 presents a crucial problem for classification known as discriminative dictionary learning. This chapter covers algorithms like label consistent K-SVD and task driven dictionary learning which are followed by proposed modifications in LC-KSVD and a proposed approach for face recognition. Finally, chapter 7 concludes the work presented in this thesis.

CHAPTER 2

PRELIMINARIES

2.1 Notations

This chapter covers some fundamental principles and key ideas that will underlie much of the ideas developed in this thesis. Throughout this thesis, signals are treated as real valued functions defined over discrete domain and generally finite. In general, lower boldcase letters (e.g. $\mathbf{x}, \mathbf{y}$) are used to denote signals. Upper case boldface letters are used to denote matrices, such as $\mathbf{X}, \mathbf{A}, \mathbf{\Phi}$ and unbolded letters (e.g., A, λ, l) are used to denote scalars. Additional conventions will be specified as needed.

2.2 l_p Norms

A norm assigns strictly positive length or size to a vector. A norm is a function $f : \mathbb{R}^n \to \mathbb{R}$ that satisfies following properties

1. $f(\mathbf{x}) > 0 \quad \forall\, \mathbf{x} > 0$
2. $f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y}) \quad \forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$
3. $f(a \cdot \mathbf{x}) = a \cdot f(\mathbf{x}) \quad \forall\, a \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^n$
4. $f(\mathbf{x}) = 0 \quad$ iff $\quad \mathbf{x} = 0$

For any integer $p \geq 1$, the l_p norm of a vector $\mathbf{x}$ is defined as

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \tag{2.1}$$

The most commonly used norm is the l_2 norm which is also known as the Euclidean norm

$$\|\mathbf{x}\|_2 = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{1/2} \tag{2.2}$$

One more important norm is the ∞-norm that gives the peak value

$$\|\mathbf{x}\|_\infty = \max_i |x|_i \tag{2.3}$$

The l_0 norm of a vector gives the number of non-zero entries in the vector. It is known as quasi norm since it doesn't satisfy property 3.

Analogous to the l_p norm of vectors, norms for matrix data can also be defined. The Frobenius norm of a matrix $\mathbf{X} \in \mathbb{R}^{n \times k}$ is defined as

$$\|\mathbf{X}\|_F = \left(\sum_{i=1}^{n} \sum_{j=1}^{k} |x_{ij}|^2 \right)^{1/2} \tag{2.4}$$

2.3 Uniqueness of Solution

In previous chapter, the problem of sparse representation was formulated. It was stated that (1.1) has infinitely many solutions if $\mathbf{D} \in \mathbb{R}^{n \times k}$ is a full rank matrix and $n < k$. To find a unique solution, additional constraints are added and equation takes the form of (1.5). The case when $f(\mathbf{x})$ computes the number of non-zero entries, it becomes a sparse representation problem. Formally, it is described as (1.8).

At this point, two important questions arise

1. Is it possible to solve P_0 uniquely?
2. If yes, under what conditions?

In this section, some theorems and definitions are presented to prove that it is possible to find a unique solution for P_0.

2.3.1 Definitions

Definition 2.1. *The spark of a matrix is defined as the smallest number of columns in the matrix that are linearly independent.*

The spark of a matrix is similar to the rank which is defined as the maximum number of independent columns in a matrix. But it is more difficult to calculate. This is because the calculation

of the spark involves an exhaustive search over all the possible combinations of vectors. By definition of spark, vectors $\mathbf{x}$ that are in the null-space of $\mathbf{D}$ must satisfy $\|\mathbf{x}\|_0 > spark(\mathbf{D})$. This means that any solution with sparsity less than or equal to the spark is unique for that value of sparsity. Spark gives a simple criterion for uniqueness which is discussed in the next subsection.

Definition 2.2. *The mutual coherence of a matrix is defined as the largest absolute inner product between different columns from* $\mathbf{D}$*.*

$$\mu(\mathbf{D}) = \max_{1 \leq i,\, j \leq k;\, i \neq j} \frac{\left|\mathbf{d}_i^T \mathbf{d}_j\right|}{\|\mathbf{d}_i\|_2 \|\mathbf{d}_j\|_2} \tag{2.5}$$

Computation of spark, being a combinatorial problem, is very difficult to solve. So another uniqueness criterion was proposed based on the mutual coherence of a matrix. Unitary matrices have zero mutual coherence. A matrix is desired to have the smallest possible value of mutual coherence so as to get as close to unitary matrices as possible.
For a dictionary $\mathbf{D} \in \mathbb{R}^{n \times 2n}$ of the form $\mathbf{D} = [\mathbf{\Phi} \quad \mathbf{\Psi}]$ where $\mathbf{\Phi}$ and $\mathbf{\Psi}$ are unitary matrices, the mutual coherence satisfies the relation $1/\sqrt{n} \leq \mu(\mathbf{D}) \leq 1$. In general, the mutual coherence of a full rank matrix of size $n \times k$ is bounded from below by

$$\mu \geq \sqrt{\frac{k-n}{k(n-1)}} \tag{2.6}$$

The equality is obtained for a family of matrices known as Grassmanian frames. For this family, spark attains the highest possible value $\mu(\mathbf{D}) = k + 1$.

2.3.2 Theorems

Theorem 2.1. *(see [16]) If a system of equations* $\mathbf{y} = \mathbf{D}\mathbf{x}$ *has a solution that satisfies* $\|\mathbf{x}\|_0 < spark(\mathbf{D})/2$*, this solution is necessarily the sparsest possible.*

Proof. Consider a solution $\mathbf{z}$ that satisfies $\mathbf{y} = \mathbf{D}\mathbf{z}$. It implies that $\mathbf{D}(\mathbf{x} - \mathbf{z}) = 0$ or $(\mathbf{x} - \mathbf{z})$ is in the null space of $\mathbf{D}$. By definition of spark

$$\|\mathbf{x}\|_0 + \|\mathbf{z}\|_0 \geq \|\mathbf{x} - \mathbf{z}\|_0 > spark(\mathbf{D}) \tag{2.7}$$

It means that if $\|\mathbf{x}\|_0 < spark(\mathbf{D})/2$, all other solutions $\mathbf{z}$ will have more number of non-zero entries than $spark(\mathbf{D})/2$. It proves that $\mathbf{x}$ is the sparsest possible solution. □

Lemma 2.1. *(see [16]) For any matrix* $\mathbf{D} \in \mathbb{R}^{n \times k}$*, following relationship is always true*

$$spark(\mathbf{A}) \geq 1 + \frac{1}{\mu(\mathbf{D})} \tag{2.8}$$

Proof. First, normalize the columns of $\mathbf{D}$ to have unit l_2 norm. Normalization does not affect spark or mutual coherence of a matrix. By definition of mutual coherence, the entries of Gram matrix $\mathbf{G} = \mathbf{D}^T\mathbf{D}$ satisfy following relations

$$\{g_{i,i} = 1 : 1 \leq i \leq k\} \quad \text{and} \quad \{g_{i,j} < \mu : 1 \leq i, j \leq k;\ i \neq j\} \tag{2.9}$$

From Gershgorin disk theorem, if a submatrix formed by taking any p columns from $\mathbf{D}$ is diagonally dominant, this submatrix is positive definite. It implies that columns of this submatrix are linearly independent. For this matrix to be positive definite, the condition $p < 1 + 1/\mu$ must hold. This condition implies $spark(\mathbf{D}) \geq p + 1 \geq 1 + 1/\mu$. □

Theorem 2.2. *(see [16]) If a system of equation* $\mathbf{y} = \mathbf{D}\mathbf{x}$ *has a solution that satisfies* $\|\mathbf{x}\|_0 < \frac{1}{2}(1 + 1/\mu(\mathbf{D}))$*, this solution is necessarily the sparsest one.*

Proof. Using lemma 2.1 and theorem 2.1, it can be seen that this theorem is indeed true. This theorem uses only a lower bound on spark. Since minimum value of coherence is $1/\sqrt{n}$, this theorem gives cardinality bound of $\sqrt{n}/2$. □

CHAPTER 3

SPARSE REPRESENTATION

Sparse representation has been an active area of research in recent years. It is a very powerful method that is used in the compression of signals. It aims to find a representation of a signal over a basis such that the representation has very few non-zero coefficients. There are two important parts to the process of finding a suitable representation

1. An overcomplete dictionary
2. Algorithms to find a linear combination of dictionary atoms and corresponding coefficients that approximate the input signal.

Many audio and image signals assume sparse representations over standard bases such as Fourier and wavelet. Researchers have devised algorithms to learn dictionaries that are best suited for a given set of input signals. These algorithms are known as *dictionary learning algorithms*.

In this chapter, we start by formulating the sparse representation problem and discuss various algorithms to solve this problem. Dictionary learning problem is discussed in subsequent chapters.

3.1 Problem Formulation

Algorithms for sparse representation aim to find an equivalent representation of a signal over a basis such that the number of non-zero coefficients is very less compared to the signal length.

Formally, this problem can be described as

$$P_0 : \quad \mathbf{y} = \mathbf{D}\mathbf{x} \quad \text{subject to} \quad \|\mathbf{x}\|_0 \leq k_0 \tag{3.1}$$

Here vector $\mathbf{y}$ represents the original signal and $\mathbf{x}$ is the coefficient vector over the dictionary $\mathbf{D}$. This problem is NP hard as it requires an exhaustive search over all possible combinations of dictionary

atoms.

In many cases, it is not possible to find the exact solution of (3.1). In such cases, this problem can be relaxed as

$$P_0 : \quad \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \quad \text{subject to} \quad \|\mathbf{x}\|_0 \leq k_0 \tag{3.2}$$

3.2 Sparse Decomposition Algorithms

It has already been pointed out that the problem described in (3.1) is NP hard and can't be solved in polynomial time. Over the years, various algorithms have been proposed to find approximate solutions. This section intends to take a closer look at some of these algorithms.

3.2.1 Convex Relaxation Techniques

One way to tackle P_0 is by replacing the l_0 norm by a smooth penalty function. An example of such approach is the FOCUSS method [17] which uses the l_p norm for fixed $p \in (0, 1]$. Another method replaces the l_0 norm with the l_1 norm. It is known as basis pursuit.

Basis Pursuit

Basis pursuit finds a representation over the dictionary such that the l_1 norm of the representation is minimized [18]. This problem can be represented as

$$P_1 : \quad \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{y} = \mathbf{D}\mathbf{x} \tag{3.3}$$

This problem is a convex optimization problem and can be solved in polynomial time. It has been shown in subsequent sections that if $\mathbf{x}$ is sparse enough then the solution of (3.3) is the same as the solution of (3.1). This problem can be relaxed by adding some margin of error.

$$P_1 : \quad \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \leq \epsilon \tag{3.4}$$

Both the problems described above can be reduced to linear programming problems [19]. Suppose $\mathbf{x} = \mathbf{u} - \mathbf{v}$ where $\mathbf{u}$ takes all the positive entries of $\mathbf{x}$ and $\mathbf{v}$ takes all the negative entries with all other entries of $\mathbf{u}$ and $\mathbf{v}$ zero. If $\mathbf{z} = [\mathbf{u}^T, \mathbf{v}^T]^T$, then it can be seen that $\|\mathbf{x}\|_1 = \mathbf{1}^T\mathbf{z}$ and $\mathbf{D}\mathbf{x} = \mathbf{D}(\mathbf{u} - \mathbf{v}) = [\mathbf{D}, -\mathbf{D}]\mathbf{z}$. With this modification P_1 problem can be posed as

$$\arg\min_{\mathbf{z}} \mathbf{1}^T\mathbf{z} \quad \text{subject to} \quad \mathbf{y} = [\mathbf{D}, -\mathbf{D}]\mathbf{z} \text{ and } \mathbf{z} > \mathbf{0} \tag{3.5}$$

This problem is a classical linear programming (LP) problem. Details can be found in [2].

LASSO Regression

LASSO, also known as least absolute shrinkage and selection operator, is a method for estimation in linear models [20]. It minimizes the sum of residuals subject to the l_1 norm of coefficient vector being less than a constant.

$$\arg\min_{\mathbf{x}} \ \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_1 \leq k_0 \tag{3.6}$$

In Lagrangian form, it can also be written as

$$\arg\min_{\mathbf{x}} \ \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda \ \|\mathbf{x}\|_1 \right\} \tag{3.7}$$

This is a quadratic programming problem and can be tackled by standard numerical analysis. But a better approach would be to use least angle regression.

LARS Algorithm

LARS, also known as least angle regression, is relatively new algorithm and a very efficient approach to solve the LASSO [20]. At the first step, LARS selects a column most correlated with the residual. Rather than fit this column completely, LARS moves the coefficient of this column continuously to its least square value. As soon as another column produces the same correlation as the first column, this column joins the process and their coefficients are moved together towards their optimal values. This process continues until all the columns join the active set.

Algorithm 1: Least Angle Regression

Data: Observation vector $\mathbf{y}$, dictionary $\mathbf{D} \in \mathbb{R}^{n \times k}$ and stopping criterion if any

Initialization: $t := 1$

1. Find the column $\mathbf{d}_j$ that is most correlated with the residual $\mathbf{r}$.
2. Move x_j from 0 to its least square coefficient $\langle \mathbf{d}_j, \mathbf{r} \rangle$, until some other column $\mathbf{d}_k$ has as much correlation with the current residual as does $\mathbf{d}_j$.
3. Move x_j and x_k in the direction defined by their least square coefficients, until some other column $\mathbf{d}_l$ has as much correlation with the residual.
4. Continue this process until all entries of $\mathbf{x}$ have been entered. After $\min(n-1, k)$ steps, a full least square solution is obtained.

A simple modification gives the LASSO path. The modification needed is that if a non-zero coefficient hits zero, this coefficient is removed from the active set and joint direction is recomputed.

3.2.2 Greedy Methods

The Core Idea

Suppose $spark(\mathbf{D}) > 2$ and the solution uses only one dictionary atom, then this column can be found as the atom that best represents $\mathbf{y}$. It takes k tests to solve this problem. The j^{th} test can be done by solving $\arg\min_{z_j} \|\mathbf{y} - z_j\mathbf{d}_j\|$ which leads to $z_j = \mathbf{d}_j^T\mathbf{y}/\|\mathbf{d}_j\|_2^2$. If $\epsilon_j = \|\mathbf{y} - z_j\mathbf{d}_j\|$ is zero, the proper solution is found. This takes $O(nk)$ time.

Greedy algorithms use this approach to find the sparse solutions. These algorithms solve the problem iteratively staring with $\mathbf{x}_0 = 0$. They maintain a set of active dictionary atoms and at each step i, an atom that is most correlated with the residual is added to this set. These atoms are used to approximate $\mathbf{y}$ and then $\mathbf{x}_I$ is calculated. This newly calculated $\mathbf{x}_I$ computes the new residual. The algorithm terminates if this residual falls below a specific threshold, otherwise algorithm goes to the next iteration.

Matching Pursuit

Matching pursuit (MP) [6] utilizes the same idea discussed above. At each iteration, it selects a column that is most correlated with the residual of previous iteration. While computing new approximant, it uses only the column selected in the current iteration.

Algorithm 2: Matching Pursuit

Data: Observation vector $\mathbf{y}$, dictionary $\mathbf{D} \in \mathbb{R}^{n\times k}$ and stopping criterion if any
Initialization: $I := 1$, $\mathbf{r}_0 = \mathbf{y}$, $S_0 = \phi$
Repeat following steps until stopping criterion is met

1. For all $j \notin S_{I-1}$ compute z_j that minimizes

$$\epsilon_j = \arg\min_{z_j} \left\|\mathbf{d}_j z_j - \mathbf{r}^{I-1}\right\|_2^2$$

   which leads to the optimal solution $z_j = \mathbf{d}_j^T\mathbf{r}^{I-1}/\|\mathbf{d}_j\|_2^2$
2. Find the minimizer j_0 such that $\epsilon_{j_0} \leq \epsilon_j \ \forall\ j$ and update $S_I = S_{I-1} \cup \{j_0\}$.
3. Set $\mathbf{x}^I = \mathbf{x}^{I-1}$ and update $x^I(j_0) = z_{j_0}$.
4. Update the residual as $\mathbf{r}^I = \mathbf{r}^{I-1} - z_{j_0}\mathbf{d}_{j_0}$.
5. $I = I + 1$.

Orthogonal Matching Pursuit (OMP)

Orthogonal matching pursuit [6], [21] is an extension of MP. At each step, OMP selects the dictionary element best correlated with the residual part of the signal. Then it produces new approximant by

projecting the signal onto the dictionary elements that have already been selected. This extension significantly improves the performance of the algorithm.

Algorithm 3: Orthogonal Matching Pursuit

Data: Observation vector $\mathbf{y}$, dictionary $\mathbf{D}$ and stopping criterion if any
Initialization: $I := 1$, $\mathbf{r}_0 = \mathbf{y}$ and $S_0 = \phi$
Repeat following steps until stopping criterion is met

1. For all columns $\mathbf{d}_j$ compute x_j that minimizes

$$\epsilon_j = \arg\min_{x_j} \left\| \mathbf{d}_j x_j - \mathbf{r}^{I-1} \right\|_2^2$$

   which leads to the optimal solution

$$x_j = \frac{\mathbf{d}_j^T \mathbf{r}^{I-1}}{\|\mathbf{d}_j\|_2^2}$$

2. Find the minimizer j_0 such that $\epsilon_{j_0} \leq \epsilon_j \; \forall \; j$ and update $S_I = S_{I-1} \cup \{j_0\}$.
3. Update provisional solution $\mathbf{x}^I$ by minimizing $\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2$ subject to $support\,\{\mathbf{x}\} = S^I$
4. Update the residual $\mathbf{r}^I = \mathbf{y} - \mathbf{D}\mathbf{x}^I$
5. $I = I + 1$.

3.3 Performance of Basis Pursuit and Greedy Algorithms

In previous sections, BP and GA have been covered. Assume that $\mathbf{y} = \mathbf{D}\mathbf{x}$ has a sparse solution with k_0 non-zero entries and $k_0 < spark(\mathbf{D})/2$. One important question to consider is whether BP or MP will be able to recover the sparsest solution. This section presents the conditions under which these algorithms recover this solution.

Theorem 3.1. *(see [6], [16]) Assume that for a system* $\mathbf{y} = \mathbf{D}\mathbf{x}$*, a sparse solution* $\mathbf{x}$ *exists. If this solution obeys*

$$\|\mathbf{x}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right) \tag{3.8}$$

an orthogonal greedy algorithm (OGA) recovers the exact solution with threshold parameter $\epsilon_0 = 0$.

Proof. Assume, without loss of generality, that the solution $\mathbf{x}$ has all k_0 non-zero entries at the

beginning and in descending order of the values $|x_j| \cdot \|\mathbf{d}_j\|_2$. Thus,

$$\mathbf{y} = \mathbf{D}\mathbf{x} = \sum_{t=1}^{k_0} x_t \mathbf{d}_t \tag{3.9}$$

For the first iteration, $I = 0$, residual $\mathbf{r}^I = \mathbf{r}^0 = \mathbf{y}$. Thus, the set of computed errors are given by

$$\epsilon_j = \arg\min_{z_j} \|\mathbf{d}_j z_j - \mathbf{y}\|_2^2 = \left\| \mathbf{d}_j \frac{\mathbf{d}_j^T \mathbf{y}}{\|\mathbf{d}_j\|_2^2} - \mathbf{y} \right\|_2^2 = \|\mathbf{y}\|_2^2 - \frac{\left(\mathbf{d}_j^T \mathbf{y}\right)^2}{\|\mathbf{d}_j\|_2^2} \tag{3.10}$$

To choose $\mathbf{d}_1$, for all $i > k_0$, following condition must be satisfied

$$\left| \frac{\mathbf{d}_1^T \mathbf{y}}{\|\mathbf{d}_1\|_2} \right| > \left| \frac{\mathbf{d}_i^T \mathbf{y}}{\|\mathbf{d}_i\|_2} \right| \tag{3.11}$$

Substituting the value of $\mathbf{y}$ from equation (3.9)

$$\left| \sum_{t=1}^{k_0} x_t \frac{\mathbf{d}_1^T \mathbf{d}_t}{\|\mathbf{d}_1\|_2} \right| > \left| \sum_{t=1}^{k_0} x_t \frac{\mathbf{d}_i^T \mathbf{d}_t}{\|\mathbf{d}_i\|_2} \right| \tag{3.12}$$

In order to consider the worst-case scenario, a lower bound for the LHS and an upper bound for the RHS has to be considered. For the LHS,

$$\begin{aligned} \left| \sum_{t=1}^{k_0} x_t \frac{\mathbf{d}_1^T \mathbf{d}_t}{\|\mathbf{d}_1\|_2} \right| &\geq |x_1| \cdot \|\mathbf{d}_1\|_2 - \sum_{t=2}^{k_0} |x_t| \cdot \left| \frac{\mathbf{d}_1^T \mathbf{d}_t}{\|\mathbf{d}_1\|_2} \right| \\ &\geq |x_1| \cdot \|\mathbf{d}_1\|_2 - \sum_{t=2}^{k_0} |x_t| \cdot \|\mathbf{d}_t\|_2 \cdot \mu(\mathbf{D}) \\ &\geq |x_1| \cdot \|\mathbf{d}_1\|_2 \left(1 - \mu(\mathbf{D})(k_0 - 1)\right) \end{aligned} \tag{3.13}$$

For the RHS,

$$\begin{aligned} \left| \sum_{t=1}^{k_0} x_t \frac{\mathbf{d}_i^T \mathbf{d}_t}{\|\mathbf{d}_i\|_2} \right| &\leq \sum_{t=1}^{k_0} |x_t| \cdot \left| \frac{\mathbf{d}_i^T \mathbf{d}_t}{\|\mathbf{d}_i\|_2} \right| \\ &\leq \sum_{t=1}^{k_0} |x_t| \cdot \|\mathbf{d}_t\|_2 \cdot \mu(\mathbf{D}) \\ &\leq |x_t| \cdot \|\mathbf{d}_1\|_2 \cdot \mu(\mathbf{D}) k_0 \end{aligned} \tag{3.14}$$

Substituting these bounds in (3.12)

$$|x_1| \cdot \|\mathbf{d}_1\|_2 (1 - \mu(\mathbf{D})(k_0 - 1)) > |x_t| \cdot \|\mathbf{d}_1\|_2 \cdot \mu(\mathbf{D}) k_0 \tag{3.15}$$

which leads to

$$1 + \mu(\mathbf{D}) > 2\mu(\mathbf{D}) k_0 \implies k_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right) \tag{3.16}$$

This means that for above value of k_0, the first stage of algorithm is successful and the chosen atom must be in the correct support. The residual can be calculated as

$$\mathbf{r}^1 = \mathbf{y} - z_I \mathbf{d}_I = \sum_{t=0}^{k_0} x_t^I \mathbf{d}_t \tag{3.17}$$

This residual is orthogonal to $\mathbf{d}_I$. So in the next iteration, the same column will not be selected. Repeating the same steps, condition (3.16) guarantees that the algorithm find the correct index. This holds true for k_0 iterations and the algorithm gives the correct solution. □

Theorem 3.2. *(see [16]) Assume that for a system* $\mathbf{y} = \mathbf{D}\mathbf{x}$*, a sparse solution* $\mathbf{x}$ *exists. If this solution obeys*

$$\|\mathbf{x}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right) \tag{3.18}$$

then solution of both P_1 *and* P_0 *are the same.*

Proof. Consider the following set that contains other solutions given by BP with larger support, less l_0 norms and these solutions are atleast as good from l_1 perspective.

$$\mathcal{C} = \{\mathbf{z} \mid \mathbf{z} \neq \mathbf{x},\ \|\mathbf{z}\|_1 \leq \|\mathbf{x}\|_1,\ \|\mathbf{z}\|_0 > \|\mathbf{x}\|_0,\ \text{and } \mathbf{D}(\mathbf{z} - \mathbf{x}) = 0\} \tag{3.19}$$

Since it has already been shown that for $\|\mathbf{x}\|_0 < (1 + 1/\mu(\mathbf{D}))/2$, $\mathbf{x}$ is the sparsest possible solution, so other solutions are necessarily denser. Removing this condition from (3.19) and putting $\mathbf{e} = \mathbf{z} - \mathbf{x}$

$$\mathcal{C}_s = \{\mathbf{e} \mid \mathbf{e} \neq 0,\ \|\mathbf{e} + \mathbf{x}\|_1 - \|\mathbf{x}\|_1 \leq 0,\ \text{and } \mathbf{D}\mathbf{e} = 0\} \tag{3.20}$$

The strategy to prove this theorem is to enlarge this set and show that even the enlarged set is empty. Lets start with $\|\mathbf{e} + \mathbf{x}\|_1 - \|\mathbf{x}\|_1 \leq 0$. It can be written as

$$\|\mathbf{e} + \mathbf{x}\|_1 - \|\mathbf{x}\|_1 = \sum_{j=1}^{k_0} (|e_j + x_j| - |x_j|) + \sum_{j > k_0} |e_j| \leq 0 \tag{3.21}$$

Using $|a+b|-|b| \geq -|a|$, above condition can be relaxed as

$$-\sum_{j=1}^{k_0} |e_j| + \sum_{j>k_0} |e_j| \leq 0 \tag{3.22}$$

which can be reduced to

$$\|\mathbf{e}\|_1 - 2 \cdot \mathbf{1}_{k_0}^T \mathbf{e} \leq 0 \tag{3.23}$$

Here $\mathbf{1}_{k_0}^T \mathbf{e}$ indicates the absolute sum of the first k_0 entries of $\mathbf{e}$. Substituting it in (3.20) gives a enlarged set

$$\mathcal{C}_s \subseteq \left\{ \mathbf{e} \mid \mathbf{e} \neq 0,\ \|\mathbf{e}\|_1 - 2 \cdot \mathbf{1}_{k_0}^T \mathbf{e} \leq 0,\ \text{and } \mathbf{De} = 0 \right\} = \mathcal{C}_s^1 \tag{3.24}$$

Now consider the part $\mathbf{De} = 0$. First multiplying this by $\mathbf{D}^T$ does not change the set. Rewriting this equation as

$$-\mathbf{e} = \left(\mathbf{D}^T\mathbf{D} - \mathbf{I}\right)\mathbf{e} \tag{3.25}$$

Taking entrywise absolute values on both sides relaxes above equation as

$$|\mathbf{e}| = \left|\left(\mathbf{D}^T\mathbf{D} - \mathbf{I}\right)\mathbf{e}\right| \leq \left|\mathbf{D}^T\mathbf{D} - \mathbf{I}\right| \cdot |\mathbf{e}| \leq |\mu(\mathbf{D})\,(\mathbf{1} - \mathbf{I})| \cdot |\mathbf{e}| \tag{3.26}$$

Here $\mathbf{1}$ is a rank 1 matrix filled with ones. Now $\mathcal{C}_s^1$ can be written as

$$\mathcal{C}_s^1 \subseteq \left\{ \mathbf{e} \mid \mathbf{e} \neq 0,\ \|\mathbf{e}\|_1 - 2 \cdot \mathbf{1}_{k_0}^T \mathbf{e} \leq 0,\ \text{and } |\mathbf{e}| \leq \frac{\mu(\mathbf{D})}{1+\mu(\mathbf{D})} \mathbf{1} \cdot |\mathbf{e}| \right\} = \mathcal{C}_s^2 \tag{3.27}$$

This set is unbounded since if $\mathbf{e} \in \mathcal{C}_s^2$, the $\alpha\mathbf{e} \in \mathcal{C}_s^2$ for all $\alpha \neq 0$. Thus, restricting the quest for normalized vectors, $\|\mathbf{e}\|_1 = 1$, the new set becomes

$$\mathcal{C}_r = \left\{ \mathbf{e} \mid \ \|\mathbf{e}\|_1 = 1,\ 1 - 2 \cdot \mathbf{1}_{k_0}^T \mathbf{e} \leq 0,\ \text{and } |\mathbf{e}| \leq \frac{\mu(\mathbf{D})}{1+\mu(\mathbf{D})} \mathbf{1} \right\} \tag{3.28}$$

If energy of a vector $\mathbf{e}$ is concentrated in its first k_0 entries, the condition $1 - 2 \cdot \mathbf{1}_{k_0}^T \mathbf{e} \leq 0$ will always be satisfied. However $\|\mathbf{e}\|_1 = 1$ and $e_j \leq \mu(\mathbf{D})/(1+\mu(\mathbf{D}))$ leads to $|e_j| = \mu(\mathbf{D})/(1+\mu(\mathbf{D}))$. Thus

$$1 - 2 \cdot \mathbf{1}_{k_0}^T \mathbf{e} = 1 - 2k_0 \frac{\mu(\mathbf{D})}{1+\mu(\mathbf{D})} \leq 0 \tag{3.29}$$

which implies that if $k_0 < (1 + 1/\mu(\mathbf{D}))/2$, set $\mathcal{C}_r$ will be necessarily empty. Hence BP will give the desired solution. □

CHAPTER 4

SPARSE REPRESENTATION BASED CLASSIFICATION

In previous chapters, underlying theory of sparse representation has been covered. Various transforms such as discrete Fourier transform, wavelet transform and SVD are used to find a compact representation of a signal. They are also used to reveal certain structures of a signal which is not apparent in time domain. In additions to these fields, sparse representation has applications as diverse as denoising, feature extraction, blind source detection and patter classification.

In recent years, sparse representation has seen a lot of applications in the field of computer vision and pattern classification. Some of them are face recognition, l_1 graph learning, dictionary learning etc [22]. This chapter considers the problem of face classification which is very famous in the field of computer vision and machine intelligence. Over the years, many approaches have been proposed to tackle this problem but only in recent years, good performance has been achieved. This chapter is devoted to the sparse representation based classification. The scope of classification is limited to the frontal view with varying expressions and under different illuminations. It is also shown that this approach gives good performance with conventional feature extraction methods such as eignefaces as well as unconventional methods like randomfaces and downsampling.

4.1 Problem Formulation

Classification problem can be described as using labeled training samples from L distinct classes to correctly determine to which class a new sample belongs to. Assume that a class i has k_i training samples arranged as columns of a matrix $\mathbf{D}_i$

$$\mathbf{D}_i = [\mathbf{d}_{i,1}, \mathbf{d}_{i,2} \dots \mathbf{d}_{i,k_i}] \tag{4.1}$$

Here each column corresponds to an image or any image feature. All face images are assumed to be grayscale of size $w \times h$. Then $\mathbf{d} \in \mathbb{R}^n$ where $n = w \times h$.

4.2 Mathematical Model of Face Images

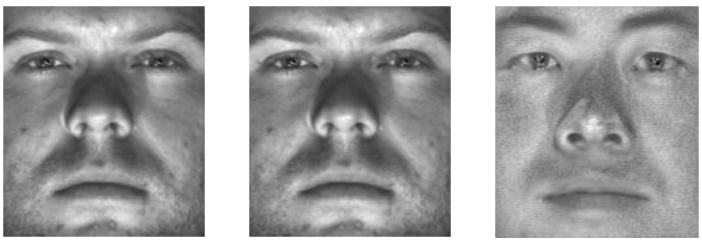
Over the years, a variety of models have been proposed to exploit the structure of $\mathbf{D}_i$ for recognition. A simple approach is to model the images from the same class as lying on a subspace [23], [24]. It has been observed that images of a person under different illuminations and with varying expressions lie on a low dimensional subspace. It implies that a test image can be written as a linear combination of training images.

$$\mathbf{y} = x_{i,1}\mathbf{d}_{i,1} + x_{i,2}\mathbf{d}_{i,2} + x_{i,3}\mathbf{d}_{i,3} + \ldots x_{i,k_i}\mathbf{d}_{i,k_i} \tag{4.2}$$

Equivalently, it can also be represented as

$$\mathbf{y} = \mathbf{D}_i\mathbf{x}_i \tag{4.3}$$

In figure 4.1, this scenario is demonstrated. 4.1b shows a reconstructed version of an image shown in 4.1a. The reconstructed version is obtained by linearly combining other images of the same person under different illuminations. Least square method is used to compute the coefficient vector $\mathbf{x}_i$. It is apparent that both the images look almost the same. There is no perceptual loss. An error of 5.37% supports this observation. Figure 4.1c shows a reconstructed version which is obtained by linearly combining images of some other person. It can be observed that this image looks nothing like the original image and gives an error of 33.51%. It is clear from these observations that this model is fairly discriminative among different classes and suitable for face classification.



(a) *(b)* *(c)*

Figure 4.1: (a) Original image (b) Image obtained by linearly combining other images of the same person under different illuminations [Error: 5.37%] (c) Image obtained by linearly combining images of some other person under different illuminations [Error: 33.51%]

4.3 Sparse Representation based Classification

4.3.1 Working of SRC

Classification problem estimates the class of test images [25]. Since class information of a test image is not known, a new matrix $\mathbf{D} \in \mathbb{R}^{n \times k}$ is defined by concatenation of matrices $\mathbf{D}_i$ defined in (4.1).

$$\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3 \dots \mathbf{D}_L] \tag{4.4}$$

A test image $\mathbf{y}$ can be expressed as a linear combination of the columns of $\mathbf{D}$ as

$$\mathbf{y} = \mathbf{D}\mathbf{x} \tag{4.5}$$

Here $\mathbf{x}$ takes the form $\mathbf{x} = [0, 0, \dots x_{i,1}, x_{i,2} \dots x_{i,k_i}, 0, 0 \dots 0]$. If the number of classes L is sufficiently large, equation (4.4) becomes a sparse representation problem. Any of the sparse recovery algorithms can be used to compute $\mathbf{x}$. In this thesis, BP, OMP, LASSO and LARS are used depending on the requirements.

Given a test image $\mathbf{y}$, first the sparse representation of $\mathbf{y}$ over $\mathbf{D}$ is computed. Ideally, all non-zero entries in $\mathbf{x}$ must be associated with the columns from a single class. In practical situations, this is not the case due to noise and errors. Nonetheless, the large coefficients are concentrated around the class which the test image belongs to. This property can be used to classify $\mathbf{y}$.

4.3.2 Classification

As stated above, the large coefficients are concentrated on the actual class of $\mathbf{y}$. Figure 4.2 shows the reconstructed coefficients for a test image of class 1. It can be noticed that the non-zero entries are indeed concentrated on the first class. Let $\delta_i : \mathbb{R}^k \to \mathbb{R}^k$ be a function that selects the coefficients associated with class i. A metric for the concentration of coefficients on the i^{th} class can be defined to classify the test image.

$$\alpha_i = \frac{\|\delta_i(\mathbf{x})\|_1}{\|\mathbf{x}\|_1} \tag{4.6}$$

If the value of α_i exceed a pre-defined threshold, label i is assigned to the test image $\mathbf{y}$. This threshold can be decided based on the concentration metric for the training images. Figure 4.3 shows the concentration of reconstructed coefficients on different face classes.

A more sophisticated approach for classification is residual based. Using the reconstructed coefficients associated with class i, the test image $\mathbf{y}$ is approximated as $\hat{\mathbf{y}}_i = \mathbf{D}\delta_i(\mathbf{x})$. Test image $\mathbf{y}$

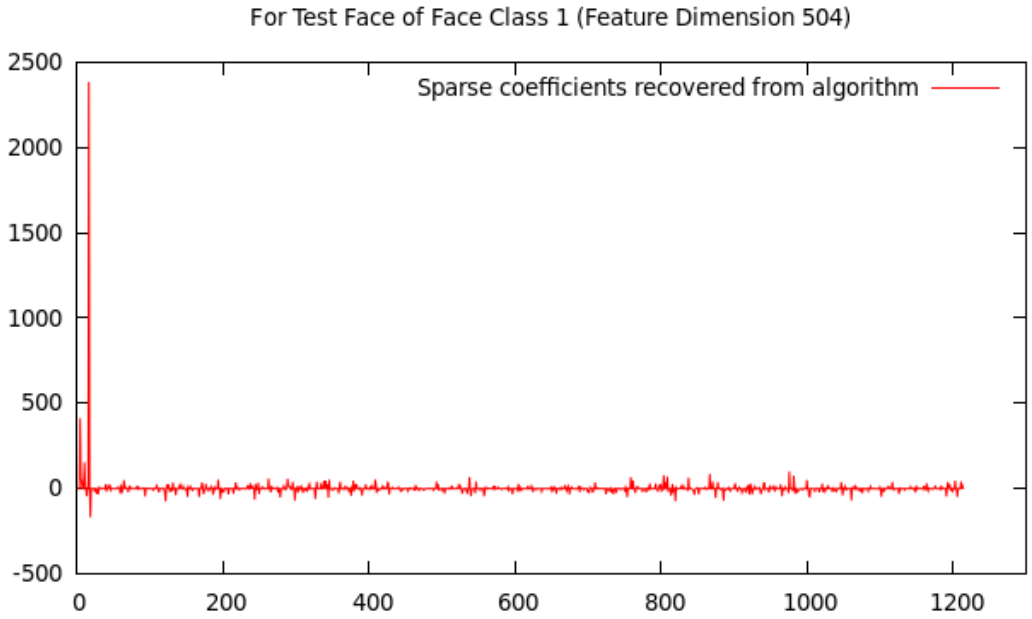


Figure 4.2: Reconstructed coefficients for a test image of class 1

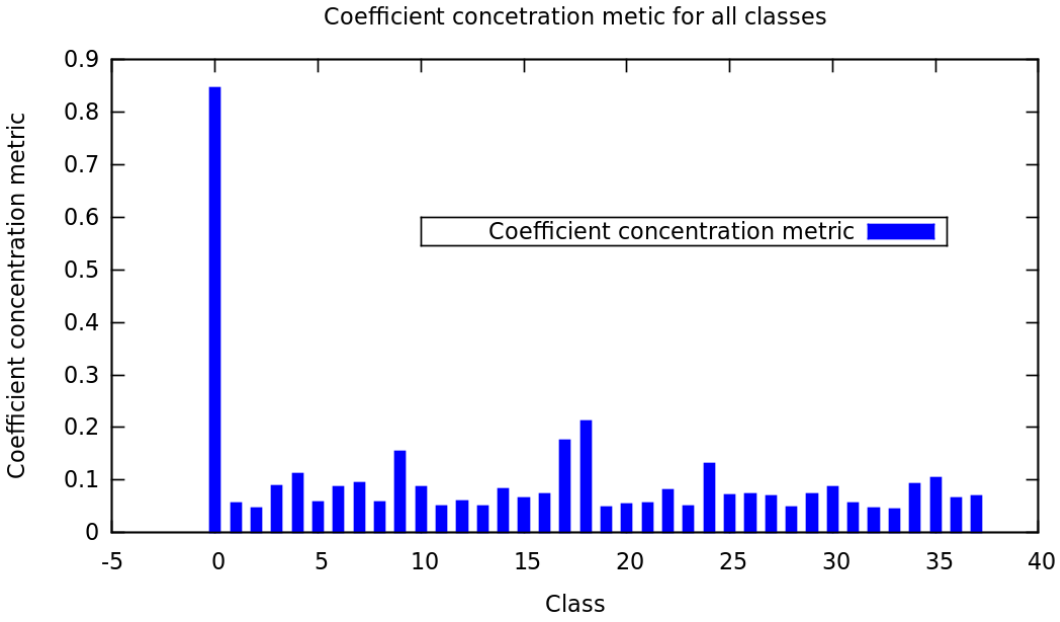


Figure 4.3: Coefficient concentration for a test image of class 1

is assigned to the class that minimizes the residual.

$$\arg\min_i \ \|\mathbf{y} - \mathbf{D}\delta_i(\mathbf{x})\|_2 \tag{4.7}$$

Figure 4.4 presents this approach for face classification. It can be noted that both the approaches give a very good degree of discrimination between the actual class of the test image and the other classes. But residual based approach is more discriminative. Hence, in this thesis latter approach is used.

4.3.3 The Algorithm

The algorithm is presented below [25].

Algorithm 4: Sparse Representation based Classification

Data: A dictionary $\mathbf{D} \in \mathbb{R}^{n \times k}$ and a test image $\mathbf{y}$

1. Normalize columns of dictionary $\mathbf{D}$ to have unit norm.
2. Compute sparse coefficient vector by solving

$$\arg\min_{\mathbf{x}} \ \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{y} = \mathbf{D}\mathbf{x}$$

3. Calculate residual corresponding to i^{th} class

$$r_i(\mathbf{y}) = \|\mathbf{y} - \mathbf{D}\delta_i(\mathbf{x})\|_2$$

4. Assign $\mathbf{y}$ to the class that minimizes $r_i(\mathbf{y})$.

4.3.4 Handling of Irrelevant Images

An important feature of a face recognition system is how it handles irrelevant images. Irrelevant images include any face image of a class that is not in the training set or an image that is not a face at all. It is desirable that these images must be discarded by the system rather than assigning them to a wrong class. Figure 4.5 shows the reconstructed coefficients corresponding to an irrelevant image. It is apparent that the coefficients are not concentrated on any particular class, rather they are scattered everywhere. This property can be used for discarding an irrelevant image.

For this purpose, a sparsity concentration index of a vector $\mathbf{x}$ is defined as

$$\text{SCI}(\mathbf{x}) = \frac{k \cdot \max_i \|\delta_i(\mathbf{x})\|_1 / \|\mathbf{x}\|_1 - 1}{k - 1} \tag{4.8}$$

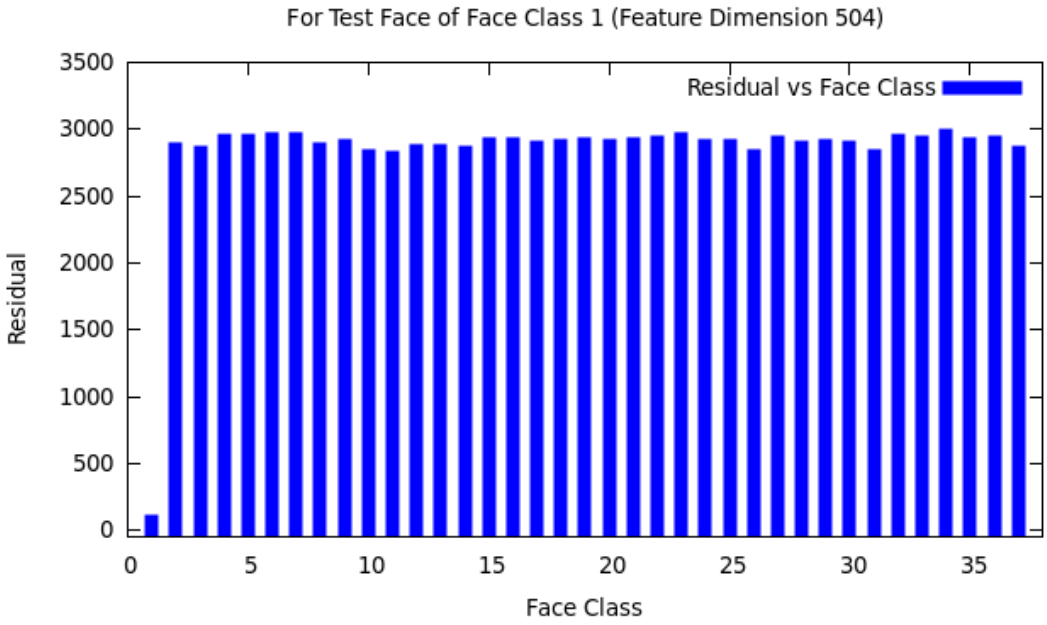


Figure 4.4: Residual for a test image of class 1

If all the non-zero coefficients are concentrated on one particular class, then $\text{SCI}(\mathbf{x}) = 1$. For any vector $\mathbf{x}$, $\text{SCI}(\mathbf{x}) \in [0, 1]$. A threshold value can be set and a test image is rejected if SCI is below that pre-defined threshold.

4.3.5 Occluded Images

In practical situations, face images might be corrupted or occluded. The person might be wearing sunglasses or face might be partially hidden by a scarf. To handle these cases, model presented in (4.5) is modified as

$$\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{e} \tag{4.9}$$

It can also be written as

$$\mathbf{y} = \begin{bmatrix} \mathbf{D} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix} \tag{4.10}$$

If $\mathbf{e}$ has a few non-zero entries i.e. a few pixels of face images are corrupted, equation (4.10) can be solved using sparse recovery algorithm and the same classification approach can be used on the $\mathbf{x}$ part of the coefficient vector.

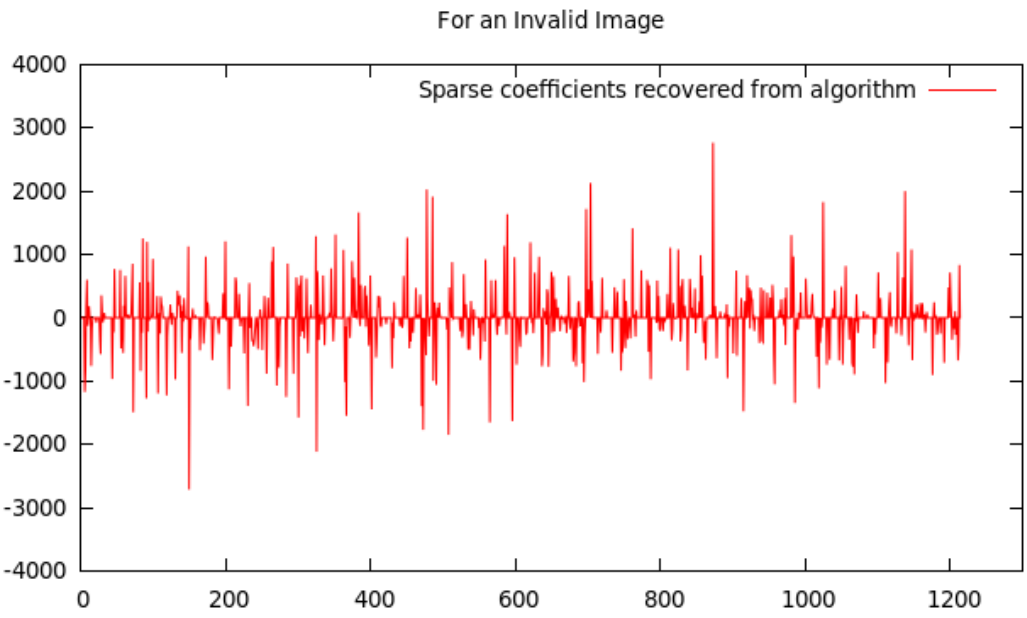


Figure 4.5: Reconstructed coefficients for an invalid image

4.4 Feature Extraction Methods

In section 4.3, dictionary $\mathbf{D}$ was said to have training images as its columns $\mathbf{d}_i$. If original images are used, size of matrix $\mathbf{D}$ becomes very large, affecting the speed of system to a great extent. A tempting solution to this problem is to extract low dimensional features from the face images and use them as the columns of $\mathbf{D}$.

Any feature extraction technique can be characterized as a linear transformation from image space to feature space. Multiplying equation (4.5) by $\mathbf{R}$ on both sides

$$\bar{y} = \mathbf{RDx} \tag{4.11}$$

In this section, three feature extraction methods are presented which are used for simulations. Figure 4.6 shows various type of features.

4.4.1 Eigenfaces

This technique is a classical method for feature extraction [26]. In this method, principal component analysis (PCA) is used for dimensionality reduction. PCA essentially aims to summarize n-dimensional vector by projecting it to a p-dimensional subspace ($p < n$). Principal components are found by minimizing the projection residual.

For calculating eigenfaces, first data is made to have zero mean. After this step, let a matrix

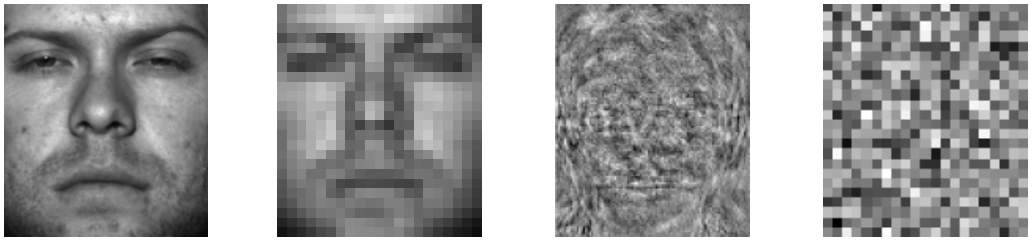
be $\mathbf{A}$ with its columns as face images $\mathbf{A} = [\Phi_1, \Phi_2, \ldots \Phi_N]$. Eigenfaces are the eigenvectors of the covariance matrix $\mathbf{C} = \mathbf{A}\mathbf{A}^T$. A detailed analysis is given in [26].

4.4.2 Downsampling

Second method used in this thesis is downsampling of the face images [25]. It is an unconventional method of feature extraction and saves the computation cost to a large extent. It'll be shown that it gives similar performance with SRC as eigenfaces.

4.4.3 Randomfaces

If entries of $\mathbf{R}$ in (4.11) are selected randomly from a zero mean normal distribution, the row vectors of $\mathbf{R}$ are referred to as randomfaces [25]. A major advantage of using randomfaces is that it is very efficient to generate a random matrix. Moreover, since $\mathbf{R}$ is independent of data, change in dataset is not an issue.



(a) Original *(b) Downsample* *(c) Eigenface* *(d) Randomface*

Figure 4.6: Various types of features

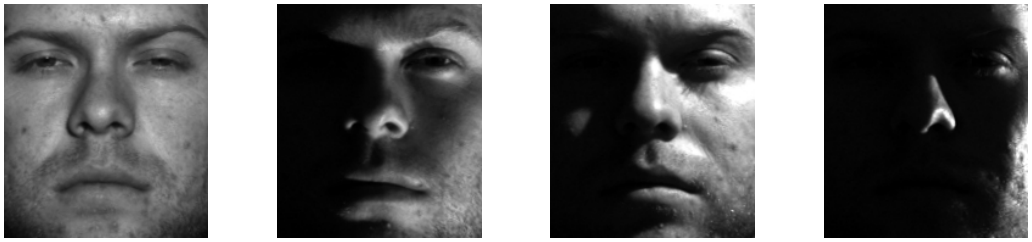
4.5 Simulations Results and Discussion

4.5.1 Datasets and Simulation Environment

All the simulation are performed in MATLAB. For all the simulations, two datasets are used.

Extended Yale B Database

This database contains 2432 images of 38 different individuals under 64 illumination conditions [27]. Randomly selected 32 images per person are used for training and rest are used for testing. Figure 4.7 shows some images of an individual .



(a) *(b)* *(c)* *(d)*

Figure 4.7: Extended Yale B database

AR Face Database

This database contains 2600 images of 100 individuals under different illuminations, varying expressions and occlusions [28]. Randomly selected 10 images per person are used for training and rest are used for testing. Figure 4.8 shows some images of a subject.

(a) *(b)* *(c)* *(d)*

Figure 4.8: AR face database

4.5.2 Recognition Rate

Recognition rate is defined as

$$\text{Recognition rate} = \frac{\text{Number of correctly classified images}}{\text{Total number of testing images}} \tag{4.12}$$

Tables 4.1 - 4.6 show the recognition rates for different feature dimensions on the extended Yale B database and the AR face database respectively. It can be observed that conventional as well as unconventional feature extraction methods perform well with SRC.

These results are compared with two classification techniques known as nearest neighbor (NN) and support vector machine (SVM). Simulation results for both NN and SVM are taken from [25]. It can be noted that SRC outperforms both the classification techniques.

Feature Dimension	30	56	120	504
Downsampling [%]	74.3	85.0	90.1	92.6
Randomfaces [%]	79.9	87.4	90.2	92.1
Eigenfaces [%]	83.9	91.7	94.2	96.2

Table 4.1: Recognition Rate of SRC on the extended Yale B database

Feature Dimension	30	56	120	504
Downsampling [%]	51.7	62.6	71.6	78.0
Randomfaces [%]	70.3	75.6	78.8	79.0
Eigenfaces [%]	74.3	81.4	85.5	88.4

Table 4.2: Recognition Rate of NN on the extended Yale B database

Feature Dimension	30	56	120	504
Downsampling [%]	48.9	69.5	79.0	91.6
Randomfaces [%]	48.8	68.6	83.4	91.4
Eigenfaces [%]	70.6	84.3	93.1	96.8

Table 4.3: Recognition Rate of SVM on the extended Yale B database

Feature Dimension	30	54	130	540
Downsampling [%]	66.4	83.5	95.8	97.3
Randomfaces [%]	56.9	78.9	91.7	92.6
Eigenfaces [%]	74.7	91.2	97.6	98.9

Table 4.4: Recognition Rate of SRC on the AR face database

Feature Dimension	30	54	130	540
Downsampling [%]	51.6	60.9	69.2	73.7
Randomfaces [%]	56.6	63.7	71.4	75.0
Eigenfaces [%]	68.1	74.8	79.3	80.5

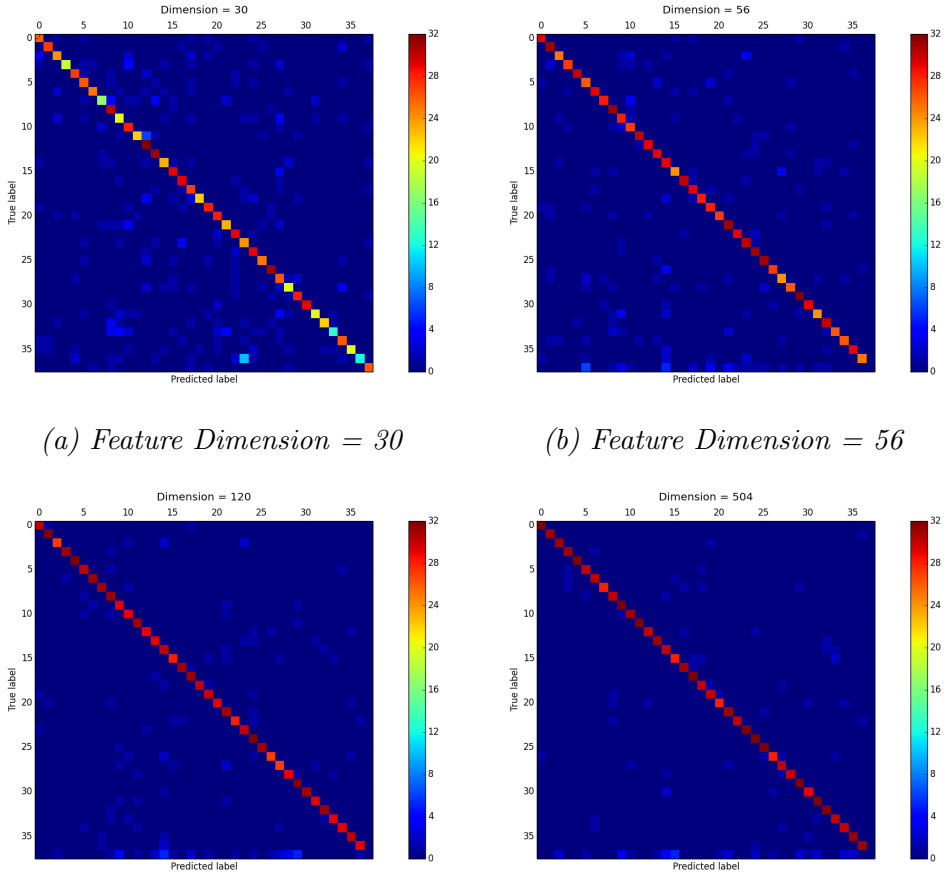
Table 4.5: Recognition Rate of NN on the AR face database

Feature Dimension	30	54	130	540
Downsampling [%]	51.4	73.0	83.4	90.3
Randomfaces [%]	54.1	70.8	81.6	88.8
Eigenfaces [%]	73.0	84.3	89.0	92.0

Table 4.6: Recognition Rate of SVM on the AR face database

4.5.3 Confusion Matrix

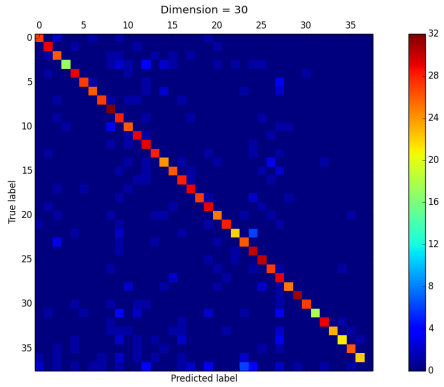
Figures 4.9 - 4.14 show the confusion matrices for SRC. Diagonal entries represent the correctly classified test images. It can be observed that these entries get brighter with the increase in feature dimension.



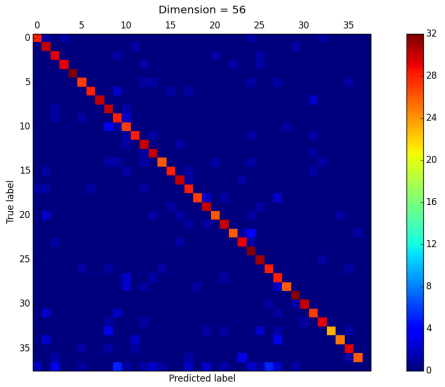
(a) Feature Dimension = 30 *(b) Feature Dimension = 56*

(c) Feature Dimension = 120 *(d) Feature Dimension = 504*

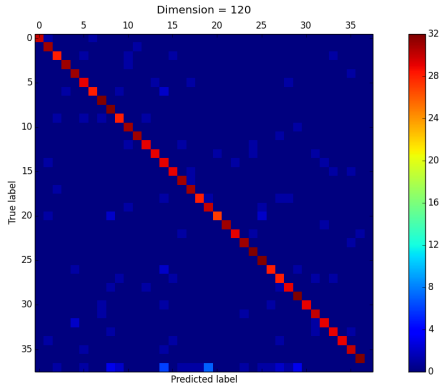
Figure 4.9: Confusion matrices of SRC with Downsampling on the extended Yale B database



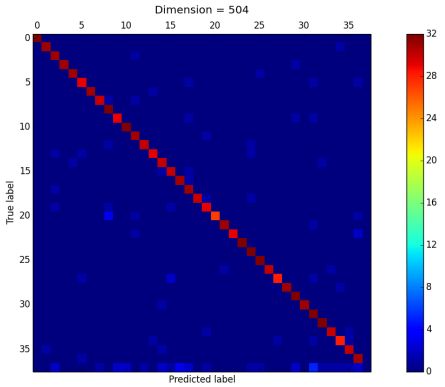
(a) Feature Dimension = 30



(b) Feature Dimension = 56

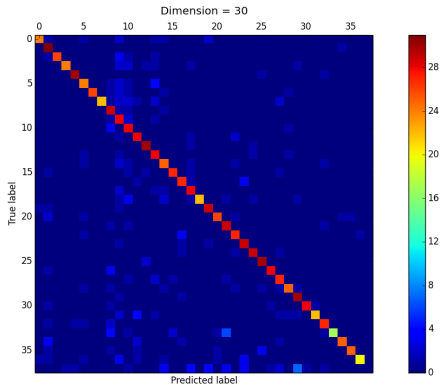


(c) Feature Dimension = 120

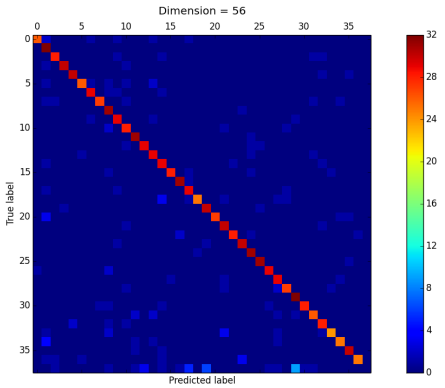


(d) Feature Dimension = 504

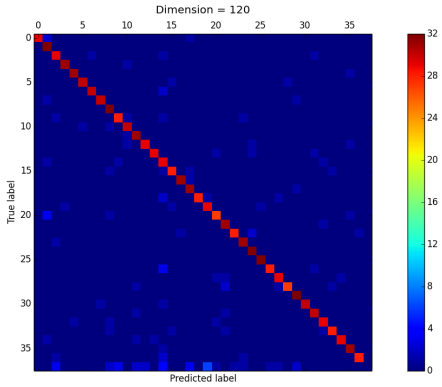
Figure 4.10: Confusion matrices of SRC with Randomfaces on the extended Yale B database



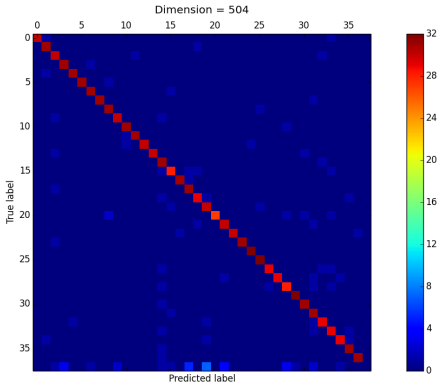
(a) Feature Dimension = 30



(b) Feature Dimension = 56

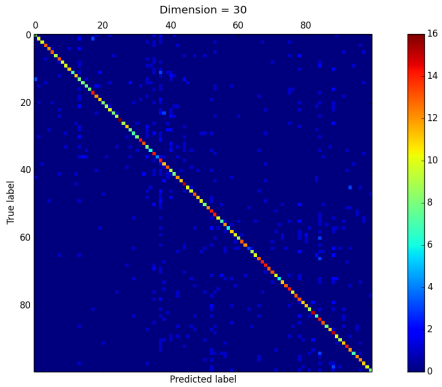


(c) Feature Dimension = 120

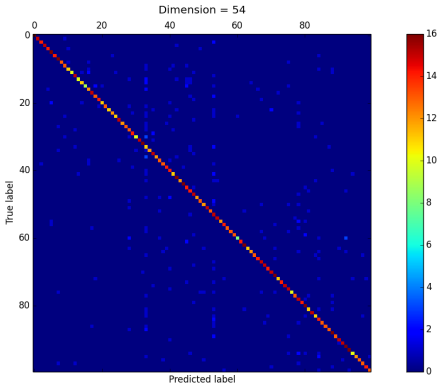


(d) Feature Dimension = 504

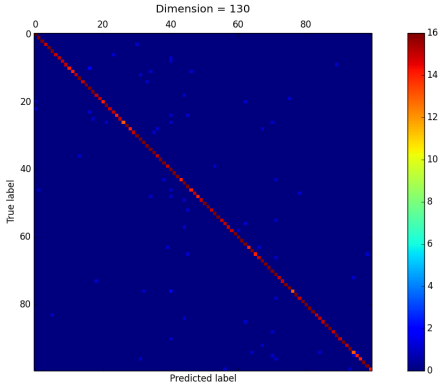
Figure 4.11: Confusion matrices of SRC with Eigenfaces on the extended Yale B database



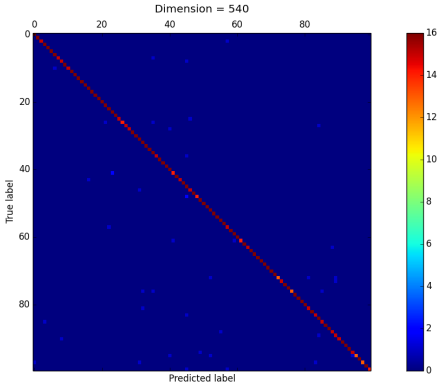
(a) Feature Dimension = 30



(b) Feature Dimension = 54

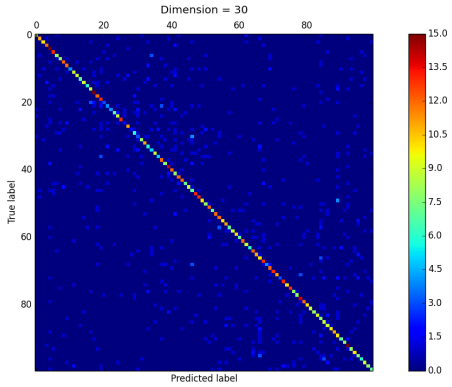


(c) Feature Dimension = 130

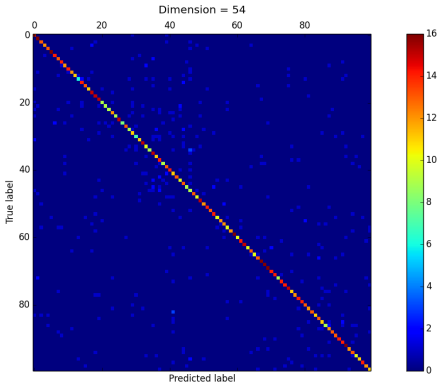


(d) Feature Dimension = 540

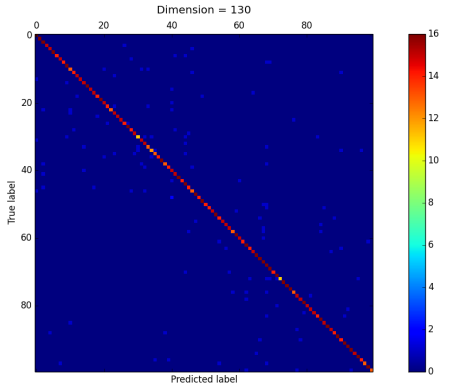
Figure 4.12: Confusion matrices of SRC with Downsampling on the AR face database



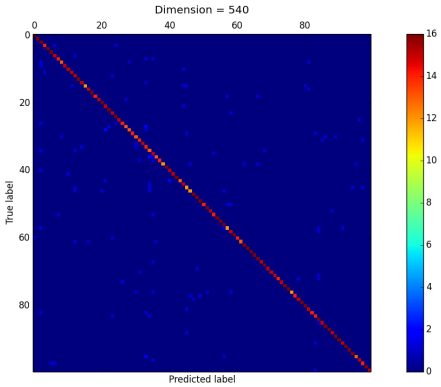
(a) Feature Dimension = 30



(b) Feature Dimension = 54

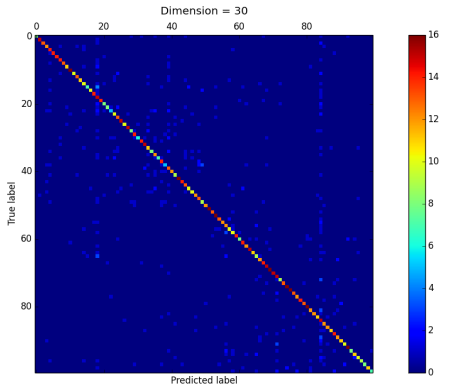


(c) Feature Dimension = 130

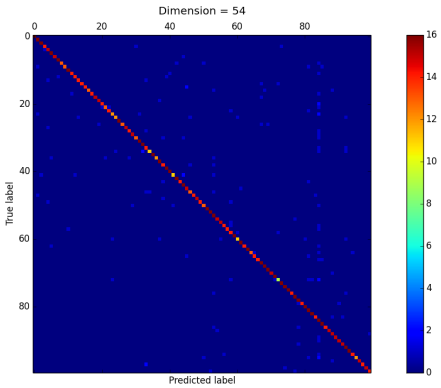


(d) Feature Dimension = 540

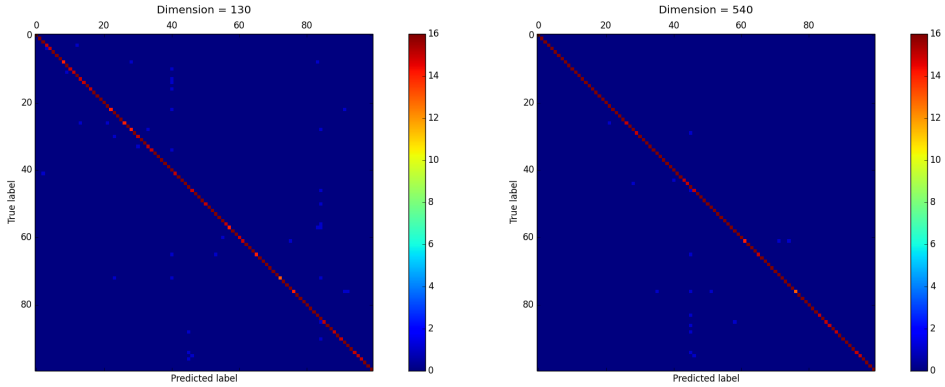
Figure 4.13: Confusion matrices of SRC with Randomfaces on the AR face database



(a) Feature Dimension = 30



(b) Feature Dimension = 54



(c) Feature Dimension = 130 *(d) Feature Dimension = 540*

Figure 4.14: Confusion matrices of SRC with Eigenfaces on the AR face database

CHAPTER 5

DICTIONARY LEARNING

In previous chapters, the main focus has been upon the sparse decomposition algorithms. It was assumed that the dictionary **D** is available as an input to these algorithms. But it turns out that the choice of these dictionaries is very crucial to the performance of sparse decomposition algorithms. For this purpose, either a pre-defined dictionary can be used which depends on the data model or a dictionary can be learnt that is best suited for the data. This chapter is devoted to the process of learning efficient dictionaries.

In chapter 3, dictionary **D** was formed by directly using the training images. Though this approach promises very good recognition rate, it becomes computationally expensive when the number of training samples is very large. This problem can also be overcome by dictionary learning. Dictionary learning makes it possible to control the size of the learned dictionary.

Over the years, various methods have been proposed for dictionary learning. These algorithms involve probabilistic methods, clustering based methods, parametric methods etc. This chapter takes a closer look at some of these algorithms.

5.1 Classical Dictionary Learning Methods

The earliest work in the field of dictionary learning involves probabilistic methods such as maximum likelihood (ML) dictionary learning, maximum a posteriori (MAP) dictionary learning etc. These algorithms iteratively solve the dictionary learning problem and use two step optimization process. Each iteration involves computing sparse coefficients for the input signals and updating the dictionary. Another approach is known as the method of optimal directions (MOD) which came after MAP and MP based algorithms and was inspired by the two step optimization used by these algorithms.

5.1.1 ML Dictionary Learning

Dictionary learning problem is modeled as [10]

$$\mathbf{Y} = \mathbf{D}\mathbf{X} + \mathbf{V} \tag{5.1}$$

Here $\mathbf{Y}$ are input samples, $\mathbf{X}$ are sparse representations over $\mathbf{D}$ and $\mathbf{V}$ are Gaussian white noise. This method seeks a dictionary $\mathbf{D}$ that maximizes the likelihood function $P(\mathbf{Y} \mid \mathbf{D})$ [11].

Two assumptions are required to solve this problem.

1. Samples are drawn independently.

$$P(\mathbf{Y} \mid \mathbf{D}) = \prod_{i=1}^{N} P(\mathbf{y}_i \mid \mathbf{D}) \tag{5.2}$$

2. $P(\mathbf{y}_i \mid \mathbf{D})$ can be written in terms of hidden variable $\mathbf{x}_i$

$$P(\mathbf{y}_i \mid \mathbf{D}) = \int P(\mathbf{y}_i, \mathbf{x}_i \mid \mathbf{D})\, d\mathbf{x_i} = \int P(\mathbf{y}_i \mid \mathbf{x}_i, \mathbf{D})\, P(\mathbf{x}_i)\, d\mathbf{x}_i \tag{5.3}$$

If $\mathbf{V}$ are Gaussian white noise vectors and $\mathbf{X}$ are zero-mean iid with Laplace distribution, the overall problem simplifies to

$$\mathbf{D} = \arg\min_{\mathbf{D}} \sum_{i=1}^{N} \min_{\mathbf{x}_i} \{\|\mathbf{D}\mathbf{x}_i - \mathbf{y}_i\| + \lambda \|\mathbf{x}_i\|_1\} \tag{5.4}$$

Since this method does not put any constraint on the entries of $\mathbf{D}$, solution will tend to increase their values in order to minimize the coefficient values. This problem is handled by putting constraints on the l_2 norm of each column of dictionary.

An iterative method was proposed to solve (5.4). It involves calculating sparse coefficients at each iteration and updating the dictionary using

$$\mathbf{D}^{(i+1)} = \mathbf{D}^{(i)} - \eta \cdot (\mathbf{D}^{(i)}\mathbf{X} - \mathbf{Y})\mathbf{X}^T \tag{5.5}$$

5.1.2 Method of Optimal Directions

The MOD method follows the K-Means outline [13]. It is an iterative algorithm with a sparse coding stage and a dictionary update stage. It has an efficient and simple way of updating the dictionary. For dictionary update at each stage, Frobenius norm of the error is minimized.

The overall mean square error can be defined as

$$\|\mathbf{E}\|_F^2 = \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \tag{5.6}$$

To minimize this error, equating the derivative of (5.6) to zero gives the dictionary update as

$$\mathbf{D}^{(i+1)} = \mathbf{Y}\mathbf{X}^{(i)^T} \cdot (\mathbf{X}^{(i)}\mathbf{X}^{(i)^T})^{-1} \tag{5.7}$$

5.1.3 Maximum A-posteriori Probability Approach

This approach attempts to merge the efficiency of the MOD method with a way to take into account preferences in the recovered dictionary [10]. But unlike ML method, it uses a-posteriori probability $P(\mathbf{D} \mid \mathbf{Y})$ [12]. Bayes' rule can be used to get likelihood expression as $P(\mathbf{Y} \mid \mathbf{D}) \propto P(\mathbf{D} \mid \mathbf{Y})P(\mathbf{D})$. Over the years, different priors are considered and corresponding dictionary update formulae are proposed. If no prior is considered, dictionary update is the same as in (5.7). Constraining the Frobenius norm of $\mathbf{D}$ to one leads to

$$\mathbf{D}^{(i+1)} = \mathbf{D}^{(i)} + \eta \cdot tr(\mathbf{X}\mathbf{E}^T\mathbf{D}^{(i)})\mathbf{D}^{(i)} \tag{5.8}$$

A major problem with this update is that several columns have a very small value of Frobenius norm and they are underused. To overcome this problem, a second constraint was proposed which constraints the columns of $\mathbf{D}$ to have unit l_2 norm. It leads to the new columnwise update

$$\mathbf{d}_j^{(i+1)} = \mathbf{d}_j^{(i)} + \eta \cdot (\mathbf{I} - \mathbf{d}_j^{(i)}\mathbf{d}_j^{(i)^T})\mathbf{E} \cdot \mathbf{x}_j^T \tag{5.9}$$

This section covered the classical methods of dictionary learning. The next section is devoted to a relatively new method known as the K-SVD algorithm which was proposed by Aharon et al.

5.2 The K-SVD Algorithm

The dictionary learning problem is formally described as

$$\arg\min_{\mathbf{D}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \leq k_0 \quad \forall\, i \tag{5.10}$$

K-SVD algorithm seeks the dictionary $\mathbf{D}$ that leads to the best possible representations $\mathbf{x}_i$. It is a generalization of the K-Means algorithm [14].

5.2.1 Working of K-SVD

K-SVD is an iterative algorithm that performs two operations at each iteration

1. Computation of the sparse coefficients over the dictionary at $(i-1)^{th}$ iteration, $\mathbf{D}^{(i-1)}$.

2. Update the dictionary using the sparse coefficients calculated in current iteration.

Sparse Coding Stage

The error term in (5.10) can be written as

$$\|\mathbf{Y} - \mathbf{DX}\|_F = \sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2 \tag{5.11}$$

Thus, sparse coding problem can be broken down to N decoupled problems as

$$\arg\min_{\mathbf{x}_i} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2 \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \leq k_0 \quad \text{for} \quad i = 1, 2, \ldots N \tag{5.12}$$

This problem can be solved by any sparse decomposition algorithm.

Dictionary Update

The dictionary update stage for this algorithm is more involved. For dictionary update, one column is updated at a time. Error defined in (5.10) can be rewritten as

$$\|\mathbf{Y} - \mathbf{DX}\|_F = \left\| \mathbf{Y} - \sum_{i=1}^{k} \mathbf{d}_i \mathbf{x}_T^i \right\|_F \tag{5.13}$$

Here the term $\mathbf{DX}$ is written as a sum of K rank-1 matrices. To update j^{th} column, equation (5.13) can be rewritten as

$$\|\mathbf{Y} - \mathbf{DX}\|_F = \left\| \mathbf{Y} - \sum_{i=1,\, i\neq j}^{k} \mathbf{d}_i \mathbf{x}_T^i - \mathbf{d}_j \mathbf{x}_T^j \right\|_F = \left\| \mathbf{E}_j - \mathbf{d}_j \mathbf{x}_T^j \right\|_F \tag{5.14}$$

The objective is to minimize this error. The error will be minimum if the second term becomes as close to $\mathbf{E}_j$ as possible. This can be used by SVD.

$$\mathbf{E}_j = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \tag{5.15}$$

Here r is the rank of $\mathbf{E}_j$. The closest rank 1 matrix (with the highest singular value) is used to replace $\mathbf{d}_j$ and $\mathbf{x}_T^j$. One issue with this update is that the new $\mathbf{x}_T^j$ is very likely to be a dense vector. A simple solution to this problem is to use only those samples $\mathbf{y}_i$ that use the current dictionary atom. The corresponding error can be described by $\mathbf{E}_j^R$. The same procedure is then applied on $\mathbf{E}_j^R$ to compute new $\mathbf{d}_j$ and $\mathbf{x}_T^j$.

5.2.2 The Algorithm

The algorithm is presented below [14].

Algorithm 5: The K-SVD Algorithm

Data: Input singals in form of a matrix $\mathbf{Y}$.
Initialization : Set the dictionary matrix $\mathbf{D}^{(0)} \in \mathbb{R}^{n \times k}$ with l_2 normalized columns. $J = 1$.
until *Stopping criterion is met*, **do**

- *Sparse coding stage*: Any sparse decomposition algorithm to compute sparse vectors $\mathbf{x}_i$ for each column $\mathbf{y}_i$ over the dictionary computed in the last iteration.
- *Codebook update*: For each column $j = 1, 2, \ldots k$ in $\mathbf{D}^{J-1}$. Update it as
  - Obtain the examples that use this atom, $\omega_j = \{i \mid 1 \leq i \leq N,\ \mathbf{x}_T^j \neq 0\}$.
  - Compute the error matrix, $\mathbf{E}_j$ by

$$\mathbf{E}_j = \mathbf{Y} - \sum_{i \neq j} \mathbf{d}_i \mathbf{x}_T^i$$

  - Restrict $\mathbf{E}_j$ by choosing columns corresponding to ω_j and obtain $\mathbf{E}_j^R$.
  - Use SVD to update dictionary column $\mathbf{d}_j$ and $\mathbf{x}_R^j$.
- $J = J + 1$

5.2.3 Some Results

This subsection shows the performance of the K-SVD algorithm. For testing

1. Generate 1000 random signals of size 64×1.
2. Learn a dictionary of size 64×500.
3. Find sparse representation of these signals over the dictionary.

Figure 5.1 shows the convergence plot of the K-SVD algorithm. It can be noted that the algorithm converges after 40 iteration and RMSE becomes < 0.02. Figure 5.2 shows a histogram plot of the l_2 norm of the residual. It can be seen that the maximum value of residual goes to 4.5%.

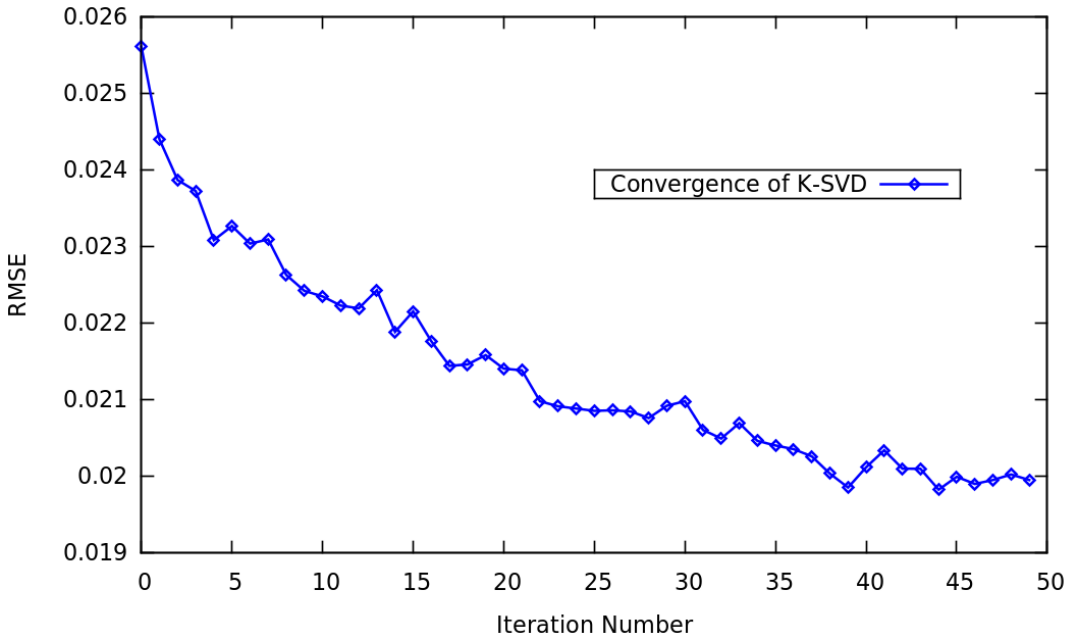


Figure 5.1: Convergence plot of the K-SVD algorithm

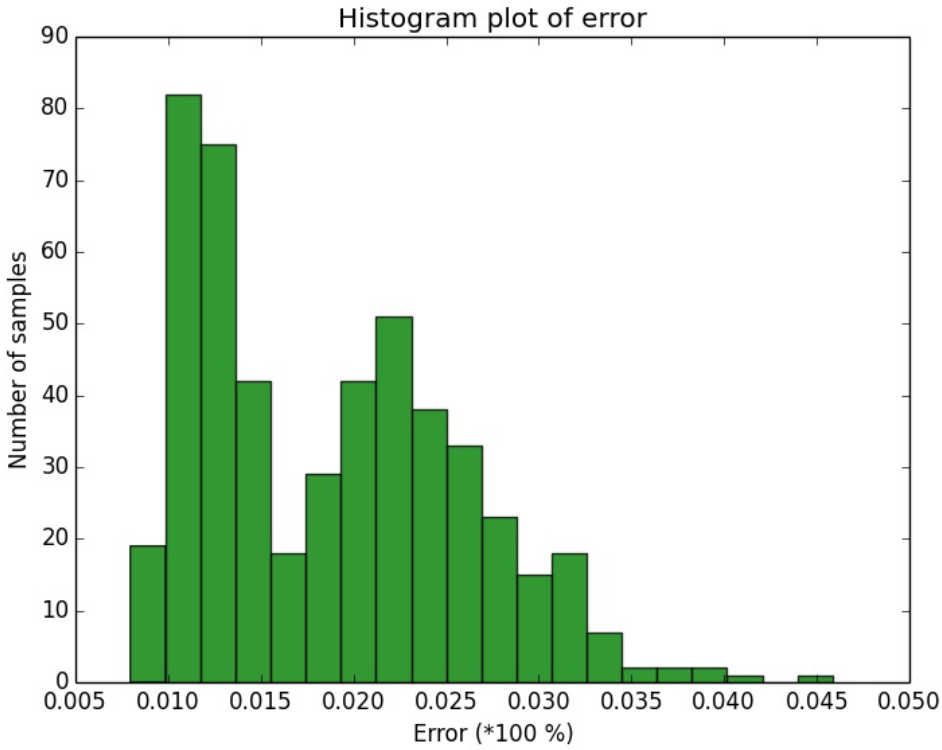


Figure 5.2: Percentage error in reconstruction

CHAPTER 6

DISCRIMINATIVE DICTIONARY LEARNING

In previous chapter, dictionary learning algorithms have been covered. The dictionaries learnt by these algorithms are reconstructive dictionaries. Reconstructive dictionaries are only concerned with just representing a signal using a few dictionary atoms. These dictionaries are no good for the classification problems since there is no constraint on the dictionary atoms used by the signals of the same class. For classification, dictionaries that can discriminate between different classes are needed. These dictionaries are known as discriminative dictionaries [29]. This chapter is devoted to the algorithms that are used to learn discriminative dictionaries.

The main focus of this chapter would be on two major algorithms known as label consistent K-SVD (LC-KSVD) and task driven dictionary learning algorithms. In subsequent sections, a modified classification approach for LC-KSVD is presented. Later in this chapter, a new approach for face classification is also presented that uses task driven dictionary learning with SRC.

6.1 Problem Formulation

As stated above, discrimination is an additional constraint when it comes to classification. The dictionary used for SRC in section 4.3 is a discriminative dictionary since ideally the sparse coefficients of the signals from a class use dictionary atoms corresponding to that class. A major problem with this dictionary is of size. When the number of training samples become very large, it becomes very costly to use such a huge dictionary for classification. An intuitive solution to this problem is to reduce the number of dictionary atoms per class. That's where discriminative dictionary learning comes into play.

Unsupervised Dictionaries

Unsupervised dictionaries only consider the residual error while learning a dictionary. They are oblivious to the class information of the input data.

Category Specific Dictionaries

To tackle the classification problem, initially it was proposed to learn dictionaries for each class separately and concatenate these individual dictionaries to form a discriminative dictionary. These type of dictionaries are known as category specific dictionaries.

Supervised Dictionaries

An intuitive solution to the classification problem is to use the label information while learning a dictionary. Such algorithms fall into the category of supervised dictionary learning. These algorithms generally solve a non-convex problem. Both LC-KSVD and task driven dictionary learning algorithms exploit the label information. These algorithms are discussed in detail in later sections.

6.2 Label Consistent K-SVD

6.2.1 The Algorithm

LC-KSVD is a discriminative dictionary learning algorithm that takes into account the label information of the data [30]. It incorporates discriminative terms into the objective function of dictionary learning. Dictionary learning problem described in (5.10) can be modified to take into account the l_1 norm to enforce sparsity.

$$\arg\min_{\mathbf{D}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F \quad \text{subject to} \quad \|\mathbf{x}_i\|_1 \leq k_0 \quad \forall\, i \tag{6.1}$$

This algorithm takes into account the structure of sparse codes over the desired dictionary. This is done by adding a label consistency term as

$$\langle \mathbf{D}, \mathbf{B}, \mathbf{X} \rangle = \arg\min_{\mathbf{D}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F + \alpha \|\mathbf{S} - \mathbf{B}\mathbf{X}\|_F \tag{6.2}$$

The second term takes into account the label consistency of reconstructed sparse codes. Here, matrix $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^N$ is referred to as discriminative sparse code matrix. The non-zero indices of $\mathbf{s}_i$

indicate the dictionary atoms that are used to represent sample $\mathbf{y}_i$. $\mathbf{S}$ takes a form of

$$\mathbf{S} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \tag{6.3}$$

It means that the first and second samples use first three dictionary atoms, the third and fourth samples use next three and so on.

6.2.2 Classification

Once the dictionary $\mathbf{D}$ is obtained, a classifier $\mathbf{W}$ is trained using ridge regression. Formally, it can be described as

$$\mathbf{W} = \arg\min_{\mathbf{W}} \|\mathbf{H} - \mathbf{W}\mathbf{X}\|_F + \lambda \|\mathbf{W}\|_F \tag{6.4}$$

which can be solved to obtain

$$\mathbf{W} = \mathbf{H}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda\, \mathbf{I})^{-1} \tag{6.5}$$

Here $\mathbf{H}$ contains the true label information of input data.

6.2.3 Solution

Equation (6.2) looks like a complex optimization problem. But it can easily be solved using K-SVD algorithm. It can be rewritten as

$$\langle \mathbf{D}, \mathbf{B}, \mathbf{X} \rangle = \arg\min_{\mathbf{D},\mathbf{B},\mathbf{X}} \left\| \begin{pmatrix} \mathbf{Y} \\ \sqrt{\alpha}\, \mathbf{S} \end{pmatrix} - \begin{pmatrix} \mathbf{D} \\ \sqrt{\alpha}\, \mathbf{B} \end{pmatrix} \mathbf{X} \right\|_F^2 \quad \text{s.t.} \quad ||\mathbf{x}_i||_0 \leq k_0 \quad 1 \leq i \leq N \tag{6.6}$$

This problem has the exact same form as (5.10) and K-SVD can be used to solve this problem.

6.2.4 Proposed Classification Approach

The proposed classification approach takes into account the parameter $\mathbf{B}$. Since optimization problem described by equation (6.2) forces the product $\mathbf{B}\mathbf{X}$ to approach $\mathbf{S}$, it can be used to classify the data.

To classify a test sample $\mathbf{y}$, the steps involved are as follows

1. Compute sparse representation $\mathbf{x}$ over the learned dictionary $\mathbf{D}$.
2. Obtain the product $\mathbf{b} = \mathbf{B}\mathbf{x}$.
3. For class i, compute concentration parameter

$$\alpha_i = \frac{\|\delta_i(\mathbf{b})\|_1}{\|\mathbf{b}\|_1} \tag{6.7}$$

4. Assign $\mathbf{y}$ to the class that maximizes α_i.

This approach gives better recognition rate as compared to the classifier defined in (6.5). It can be verified by the results presented in the next section.

6.2.5 Simulation Results and Discussion

This subsection presents the results obtained for LC-KSVD with extended Yale B database and AR face database.

Recognition Rate

Tables 6.2 and 6.1 show the comparison between recognition rate obtained by the trained classifier defined in (6.4) and the proposed classification approach for extended Yale B database and AR face database. It can be verified that the performance of SRC is superior as compared to the trained classifier for all feature dimensions. Trained classifier's performance falls drastically for low dimensions whereas SRC performs very well even for dimension as low as 56 and 120. From these results it can be concluded that the proposed approach gives better performance.

Dictionary sizes used are 760 (20 dictionary atoms per person) and 700 (7 dictionary atoms per person) for extended Yale B database and AR face database respectively.

Feature Dimension	Trained Classifier [%]	The Proposed Approach [%]
504	91.2	92.4
120	75.1	81.7
56	36.3	61.3

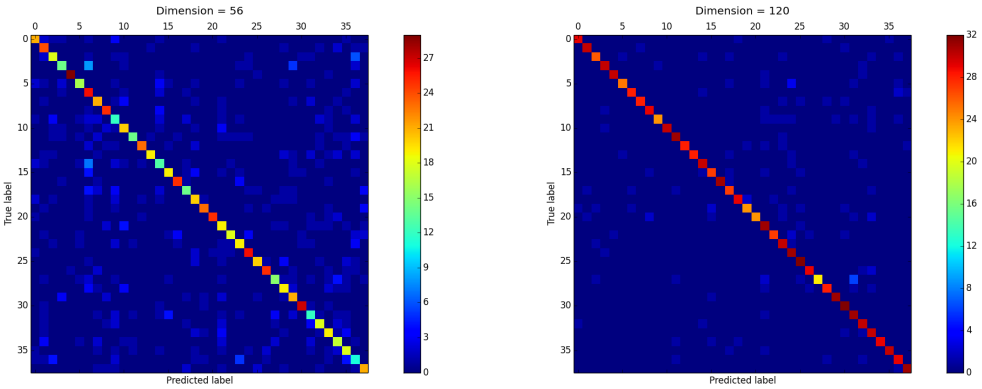
Table 6.1: Recognition Rate for LC-KSVD on the extended Yale B database

Feature Dimension	Trained Classifier [%]	The Proposed Approach [%]
540	87.7	91.8
130	67.8	74.2
54	43.8	54.6

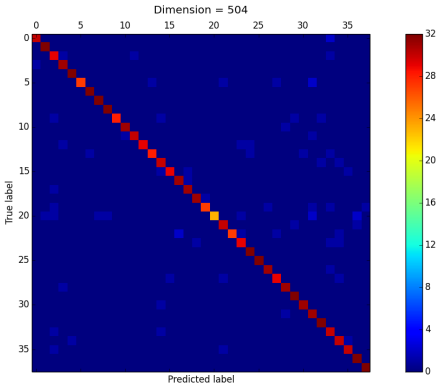
Table 6.2: Recognition Rate for LC-KSVD on the AR face database

Confusion Matrix

Figure 6.1 and 6.2 show the confusion matrices for both extended Yale B database and AR face database. It can be noted that diagonal entries take higher values with the increase in feature dimension.

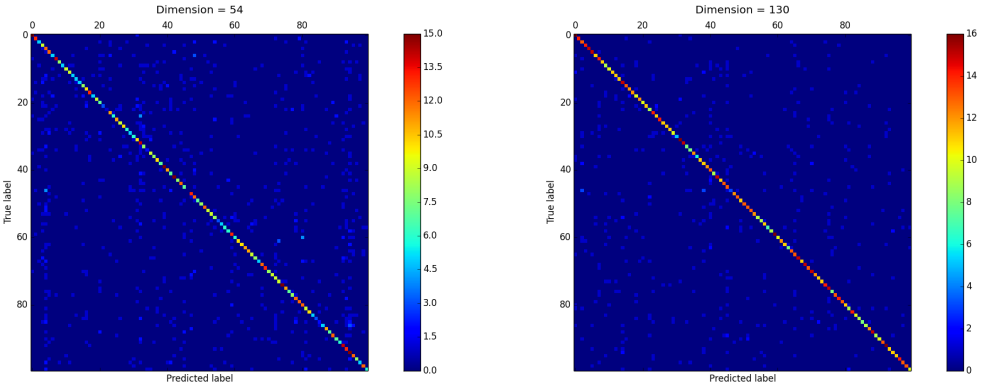


(a) Feature Dimension = 56 *(b) Feature Dimension = 120*



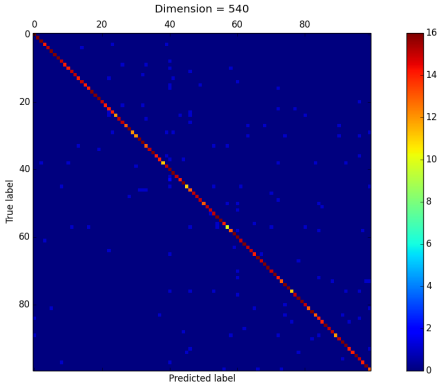
(c) Feature Dimension = 504

Figure 6.1: Confusion matrices of LC-KSVD with SRC on the AR face database



(a) Feature Dimension = 54

(b) Feature Dimension = 130



(c) Feature Dimension = 540

Figure 6.2: Confusion matrices of the LC-KSVD with SRC on the AR face database

6.3 Task Driven Dictionary Learning

6.3.1 Background

In this section, another algorithm to learn a discriminative dictionary is presented [31]. First, a generalized task driven dictionary learning algorithm is presented which is followed by the proposed approach for multiclass classification. The proposed approach uses task driven dictionary learning algorithm with SRC for face classification.

Let's assume that the training data set is represented as $(\mathbf{y}^{(i)}, z^{(i)})$ for $1 \leq i \leq N$. Here $z^{(i)}$ is true label of $\mathbf{y}^{(i)}$. Input data matrix is defined as $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N}$. Model parameters are denoted by

$\mathbf{W}$. Task driven dictionary learning can be formulated as

$$\underset{\mathbf{D},\,\mathbf{W}}{\arg\min} \quad f(\mathbf{D},\ \mathbf{W}) + \nu \left\|\mathbf{W}\right\|_F^2 \tag{6.8}$$

where f has a form

$$f(\mathbf{D},\ \mathbf{W}) = \frac{1}{N}\sum_{i=1}^{N} l_s(z_i,\ \mathbf{W},\ \mathbf{x}_i) \tag{6.9}$$

Sometimes, one is interested in minimization of the expected cost

$$f(\mathbf{D},\ \mathbf{W}) = \mathbb{E}_{\mathbf{y}}[l_s(z,\ \mathbf{W},\ \mathbf{x}(\mathbf{y},\ \mathbf{D}))] \tag{6.10}$$

In this thesis, cost defined in (6.9) is used.

To get an optimal solution of the problem defined in (6.9), gradient descent algorithm is used. If cost defined in (6.10) is used, algorithm can simply be modified by switching to stochastic gradient descent.

The next important point to consider is the differentiability of f with respect to $\mathbf{D}$ and $\mathbf{W}$.

$$\begin{aligned} \nabla_{\mathbf{W}} f(\mathbf{D},\ \mathbf{W}) &= \nabla_{\mathbf{W}} l_s(\mathbf{z},\ \mathbf{W},\ \mathbf{X}) \\ \nabla_{\mathbf{D}} f(\mathbf{D},\ \mathbf{W}) &= -\mathbf{D}\boldsymbol{\beta}^*\mathbf{X}^T + (\mathbf{Y} - \mathbf{D}\mathbf{X})\boldsymbol{\beta}^{*^T} \end{aligned} \tag{6.11}$$

Here $\boldsymbol{\beta}^*$ is a matrix formed by concatenating the vectors β^* which are defined for a signal $\mathbf{x}_i$ as

$$\beta^*_{\Lambda^C} = 0 \quad \text{and} \quad \beta^*_{\Lambda} = (\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda + \lambda_2\mathbf{I})^{-1}\nabla_{\mathbf{x}_{i_\Lambda}} l_s(z_i,\ \mathbf{W},\ \mathbf{x}_i) \tag{6.12}$$

where $\Lambda = \left\{j \mid x_{i_j} \neq 0\right\}$.

6.3.2 The Algorithm

The algorithm is on the next page. As stated earlier, discriminative dictionary leaning problems are non-convex in nature. So it is very important to initialize $\mathbf{D}$ and $\mathbf{W}$ properly, otherwise algorithm can lead to very poor results. $\mathbf{D}$ is initialized using category-specific dictionary learning and $\mathbf{W}$ is initialized using ridge regression as in (6.4).

6.3.3 The Proposed Approach

This subsection describes the proposed approach for face classification. Face classification is a multiclass classification problem and it can be solved in several ways. One simple method is to use a set of binary classifiers in a "one-vs-all" setting. This method doesn't scale well to large

Algorithm 6: Gradient Descent Algorithm for Task Driven Dictionary Learning

Data: Training samples $\mathbf{Y}$ and regularization parameters λ_1, λ_2 and ν.
Initialization: J = 1, $\mathbf{D}_0$ and $\mathbf{W}_0$
Repeat following steps for specified number of iterations

1. Sparse coding stage: For each training sample $\mathbf{y}_i$, compute sparse representations $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$.
$$\mathbf{x}_i \leftarrow \arg\min_{\mathbf{x}_i} \ \frac{1}{2}\left\|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\right\|_2^2 + \lambda_1 \left\|\mathbf{x}_i\right\|_1 + \lambda_2 \left\|\mathbf{x}_i\right\|_2^2$$
2. Form a matrix $\boldsymbol{\beta}^*$ by concatenating β^* for each signal as
   - Compute Λ as
   $$\Lambda \leftarrow \left\{j \mid x_{i_j} \neq 0\right\}$$
   - Compute β^*
   $$\beta^*_{\Lambda^C} = 0 \quad \text{and} \quad \beta^*_{\Lambda} = (\mathbf{D}_{\Lambda}^T\mathbf{D}_{\Lambda} + \lambda_2\mathbf{I})^{-1}\nabla_{\mathbf{x}_{i_\Lambda}} l_s(z,\ \mathbf{W},\ \mathbf{x}_i)$$
3. Update $\mathbf{D}$ and $\mathbf{W}$
$$\mathbf{W} \leftarrow \mathbf{W} - \rho(\nabla_{\mathbf{W}} l_s(\mathbf{z},\ \mathbf{W},\ \mathbf{X}) + \nu\mathbf{W})$$
$$\mathbf{D} \leftarrow \mathbf{D} - \rho(-\mathbf{D}\boldsymbol{\beta}^*\mathbf{X}^T + (\mathbf{Y} - \mathbf{D}\mathbf{X})\beta^{*^T})$$
4. J = J + 1

datasets. Another possibility is to use a multiclass cost function. The proposed approach for face classification uses latter since a single dictionary and a multiclass cost function is more appropriate for large datasets.

Dictionary Learning

In the setting of binary classification where $z^{(i)} \in \{0,\ 1\}$, hypothesis takes the form

$$h_{\mathbf{w}}(\mathbf{x}^{(i)}) = \frac{1}{1 + exp(-\mathbf{w}^T\mathbf{x}^{(i)})} \tag{6.13}$$

and the cost function is defined as

$$J(\mathbf{w}) = -\frac{1}{N}\left[\sum_{i=1}^{N} z^{(i)} \log h_{\mathbf{w}}(\mathbf{x}^{(i)}) + (1 - z^{(i)}) \log(1 - h_{\mathbf{w}}(\mathbf{x}^{(i)}))\right] \tag{6.14}$$

When generalizing it for the case of multiclass classification, output of the hypothesis becomes an L dimensional vector, where L is the number of classes.

$$h_{\mathbf{W}}(\mathbf{x}^{(i)}) = \begin{bmatrix} p(z = 1 \mid \mathbf{x}^{(i)}, \mathbf{W}) \\ p(z = 2 \mid \mathbf{x}^{(i)}, \mathbf{W}) \\ \vdots \\ p(z = L \mid \mathbf{x}^{(i)}, \mathbf{W}) \end{bmatrix} = \frac{1}{\sum_{j=1}^{L} e^{\mathbf{w}_j^T\mathbf{x}^{(i)}}} \begin{bmatrix} e^{\mathbf{w}_1^T\mathbf{x}^{(i)}} \\ e^{\mathbf{w}_2^T\mathbf{x}^{(i)}} \\ \vdots \\ e^{\mathbf{w}_L^T\mathbf{x}^{(i)}} \end{bmatrix} \tag{6.15}$$

In this setting, the cost function is defined as

$$J(\mathbf{W}) = -\frac{1}{N}\left[\sum_{i=1}^{N}\sum_{j=1}^{L} 1\left\{z^{(i)} = j\right\} \log \frac{e^{\mathbf{w}_j^T\mathbf{x}^{(i)}}}{\sum_{l=1}^{L} e^{\mathbf{w}_l^T\mathbf{x}^{(i)}}}\right] \tag{6.16}$$

The derivatives with respect to $\mathbf{W}$ and $\mathbf{x}$ are

$$\nabla_{\mathbf{w}_j}(\mathbf{W}) = -\frac{1}{N}\sum_{i=1}^{N}\left[\mathbf{x}^{(i)}\left(1\left\{z^{(i)} = j\right\} - \frac{e^{\mathbf{w}_j^T\mathbf{x}^{(i)}}}{\sum_{l=1}^{L} e^{\mathbf{w}_l^T\mathbf{x}^{(i)}}}\right)\right] \tag{6.17}$$

$$\nabla_{\mathbf{x}^{(i)}}(\mathbf{W}) = -\frac{1}{N}\left[\mathbf{w}_j - \frac{e^{\mathbf{w}_j^T\mathbf{x}^{(i)}}\mathbf{w}_j}{\sum_{l=1}^{L} e^{\mathbf{w}_l^T\mathbf{x}^{(i)}}}\right] \tag{6.18}$$

In equation (6.18), j is the class of sample $\mathbf{x}^{(i)}$.

Classification

For classification, SRC is used with the learned dictionary **D**. The residual method is used which has already been described in section 4.3.

6.3.4 Simulation Results and Discussion

This subsection presents the simulation results for the proposed approach for face classification.

Recognition Rate

Tables 6.3 and 6.4 show the comparison of recognition rates between the proposed approach and modified LC-KSVD algorithm for extended Yale B database and AR face database. It is apparent that this approach performs well for high dimension whereas the performance is acceptable for the dimensions as low as 30 and 56. For AR face database, performance is very good for dimensions 540 and 130 but it is not as good for lower dimensions 30 and 56. This is because AR face database contains images with varying expressions and occlusions. Downsampling these images to 6×5 results in tremendous loss of information. LC-KSVD outperforms the proposed approach only for feature dimension 504 on the extended Yale B database. Nonetheless, the performance is not very bad. Since both the techniques use SRC for classification, it can also be concluded that the task driven dictionary learning algorithm gives more discriminative dictionary than the LC-KSVD.

Dictionary sizes used are 760 (20 dictionary atoms per person) and 700 (7 dictionary atoms per person) for extended Yale B database and AR face database respectively.

Feature Dimension	LC-KSVD [%]	The Proposed Approach [%]
504	92.3	90.9
120	81.7	83.1
56	61.3	76.2
30	N/A	61.2

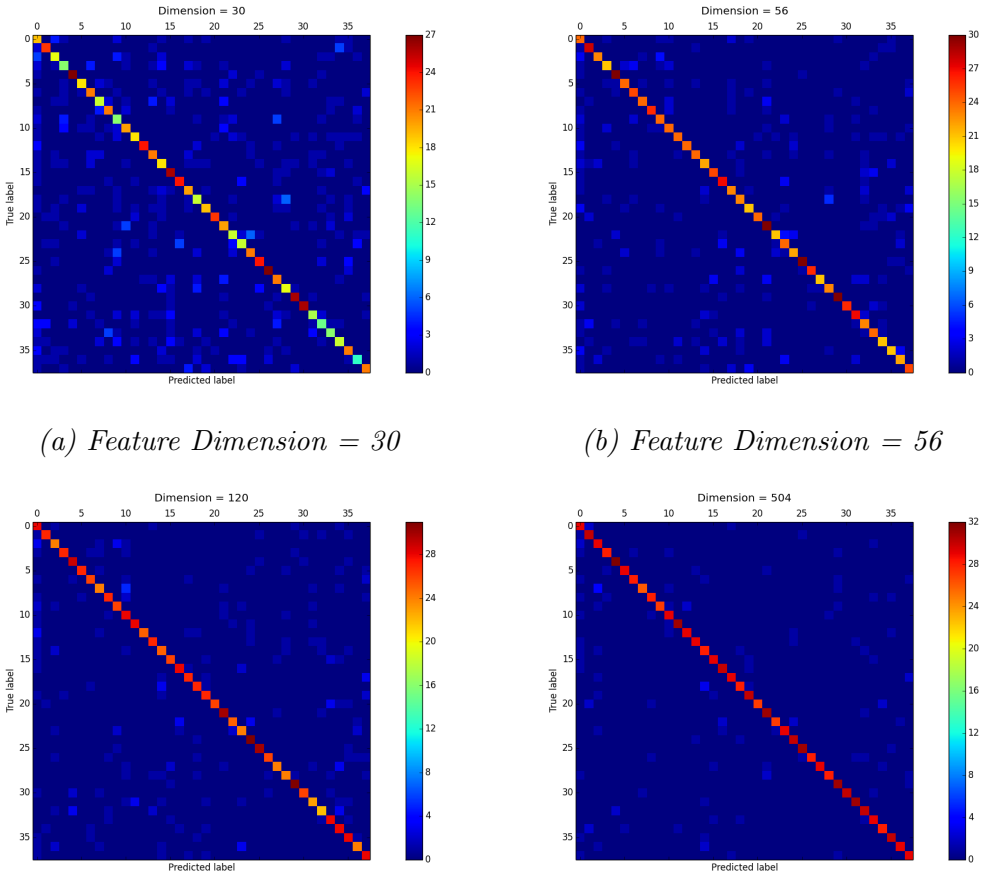
Table 6.3: Recognition Rate of the proposed approach on the extended Yale B database

Feature Dimension	LC-KSVD [%]	The Proposed Approach [%]
540	87.7	94.8
130	67.8	82.4
54	43.8	64.3
30	N/A	52.4

Table 6.4: Recognition Rate of the proposed approach on the AR face database

Confusion Matrix

Figures 6.3 and 6.4 show the confusion matrices for the proposed approach on both the databases. Here also, it can be observed that values of diagonal entries become higher with the increase in the feature dimension.



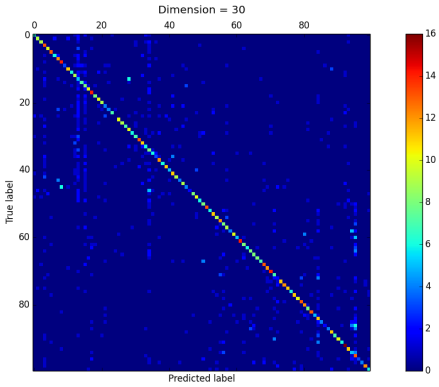
(a) Feature Dimension = 30

(b) Feature Dimension = 56

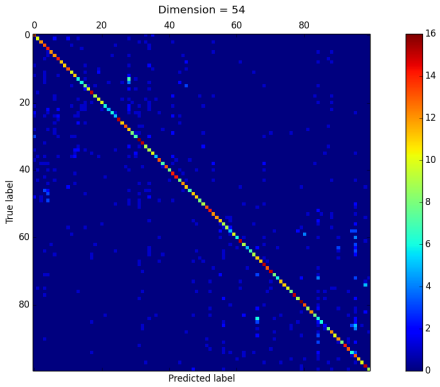
(c) Feature Dimension = 120

(d) Feature Dimension = 504

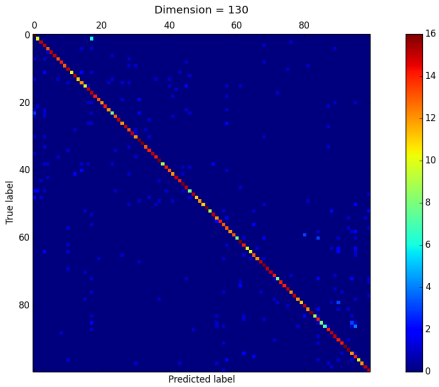
Figure 6.3: Confusion matrices of the proposed approach on the AR face database



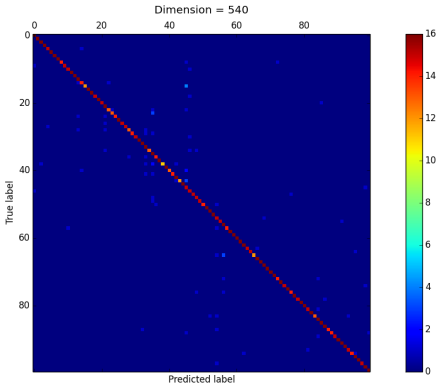
(a) Feature Dimension = 30



(b) Feature Dimension = 54



(c) Feature Dimension = 130



(d) Feature Dimension = 540

Figure 6.4: Confusion matrices of the proposed approach on the AR face database

CHAPTER 7

CONCLUSION

In chapter 1, the objectives of this thesis were

1. Study of underlying theory of sparse representation and various sparse decomposition algorithms such as orthogonal matching pursuit, basis pursuit, LASSO, LARS etc.
2. Study of various dictionary learning algorithms such as ML dictionary learning, MAP dictionary learning, method of optimal directions and the K-SVD algorithm.
3. Study of discriminative dictionary learning algorithms and their applications in the field of face recognition.

The underlying theory of sparse representation was presented in chapter 2. This chapter covered the problem of uniqueness of solution for sparse representation. In chapter 3, various algorithms for sparse decomposition were covered. This chapter covered algorithms like basis pursuit that gives the exact solution under certain conditions as well as greedy approaches like OMP that gives an approximate solution to the sparse representation problem. This concludes our first objective.

The second objective was looked at in chapter 5. Various classical as well as recent dictionary learning algorithms were presented. The extension of this problem for classification known as discriminative dictionary learning was covered in chapter 6.

Rest of the thesis covered the third objective. In chapter 4, a sparse representation based framework for classification was presented. It was verified that this framework gives very good performance with conventional as well as unconventional feature extraction techniques. In chapter 5, a proposed classification approach for face classification using LC-KSVD was presented and its performance was found to be superior than the trained classifier. It was followed by a proposed approach for face classification problem. The performance of this approach was verified on two publicly available datasets.

From the work presented in thesis, it can be concluded that sparse representation indeed provides high performance for the face classification problem. However the discriminative dictionary learning algorithms presented in this thesis don't give good performance for lower dimensions and more work is required in this direction to devise algorithms with more discriminative capabilities.

REFERENCES

[1] G. Strang, *Introduction to linear algebra*. SIAM, 2003.

[2] M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer, 2010.

[3] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.

[4] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via 1 minimization," *Proceedings of the National Academy of Sciences*, vol. 100, no. 5, pp. 2197–2202, 2003.

[5] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constructive approximation*, vol. 13, no. 1, pp. 57–98, 1997.

[6] J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[7] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[8] D. Needell, J. Tropp, and R. Vershynin, "Greedy signal recovery review," in *42nd Asilomar Conference on Signals, Systems and Computers*, IEEE, 2008, pp. 1048–1050.

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

[10] I. Tosic and P. Frossard, "Dictionary learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, 2011.

[11] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Neural computation*, vol. 12, no. 2, pp. 337–365, 2000.

[12] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural computation*, vol. 15, no. 2, pp. 349–396, 2003.

[13] K. Engan, S. O. Aase, and J. Husoy, "Frame based signal compression using method of optimal directions (mod)," in *IEEE International Symposium on Circuits and Systems, 1999. ISCAS'99. Proceedings of the 1999*, IEEE, vol. 4, 1999, pp. 1–4.

[14] 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, 2006.

[15] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[16] 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.

[17] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.

[18] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.

[19] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[20] T. Hastie, R. Tibshirani, J. Friedman, T Hastie, J Friedman, and R Tibshirani, *The elements of statistical learning*, 1. Springer, 2009, vol. 2.

[21] T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4680–4688, 2011.

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

[23] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218–233, 2003.

[24] A. S. Georghiades, P. N. Belhumeur, and D. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[25] J. Wright, A. Y. Yang, A. Ganesh, S. 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, 2009.

[26] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91.*, IEEE, 1991, pp. 586–591.

[27] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[28] A. Martinez and R Benavente, "Cvc technical report# 24," *The AR Face Database*, 1998.

[29] 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, 2008. CVPR 2008.*, IEEE, 2008, pp. 1–8.

[30] Z. Jiang, Z. Lin, and L. Davis, "Label consistent k-svd: learning a discriminative dictionary for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2651–2664, 2013, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2013.88.

[31] 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, 2012.