DESIGN OF COMPACT AND DISCRIMINATIVE DICTIONARIES

A THESIS

submitted by

SHYJU WILSON

for the award of the degree

of

DOCTOR OF PHILOSOPHY

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY HYDERABAD

December 2017

DECLARATION

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and also can evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

(Signature)

Shyju Wilson.

(Name)

CS10P006.

(Roll No.)

Approval Sheet

This thesis entitled "Design of Compact And Discriminative Dictionaries" by Mr. Shyju Wilsonis approved for the degree of Doctor of Philosophy from IIT Hyderabad.

Prof. R. K. Patney
Dept. of EE, IIT Delhi
Examiner 1

SNMerchant

Prof. Shabbir N. Merchant
Dept. of EE, IIT Bombay
Examiner 2

V.B. Vineeth

Dr. Vineeth N. Balasubramanian
Dept. of CSE, IIT Hyderabad
Internal Examiner

C. Krishna Mohan

Dr C. Krishna Mohan
Dept. of CSE, IIT Hyderabad
Adviser/Guide

Dr. Sumohana Channappayya
Dept. of EE, IIT Hyderabad
Chairman

THESIS CERTIFICATE

This is to certify that the thesis entitled **Design of compact and discriminative dictionaries** submitted by **Shyju Wilson** to Indian Institute of Technology, Hyderabad for the award of the degree of Doctor of Philosophy is a bonafide record of research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

C. Krishna Mohan

Dr. C. Krishna Mohan
Associate Professor
Dept. of Computer Science and Engg.

Hyderabad - 502285

Date: December 18, 2017

Dedicated to my parents

ACKNOWLEDGMENTS

I would like to give my sincere gratitude to Dr. C. Krishna Mohan for nurturing my research attitude and scientific temper under his guidance. His constant support, guidance, and patience helped me to overcome difficulties faced throughout my PhD days. He has created exceptional research environment which brings out the best in me to do more productive work.

I deeply thankful to our director, Prof. U.B. Desai, for creating advanced computing facilities which tremendously helped me to conduct experiments in the research work. I thank Dr. Sri Rama Murty for his guidance and encouragement during my research work. I am grateful to Dr. M. V. Panduranga Rao and Dr. Naveen Sivadasan for their valuable advise and support during my days at IIT Hyderabad. I am sincerely grateful to my doctoral committee members Dr. Sobhan Babu, Dr. Sathya Peri, Dr. M. V. Panduranga Rao and Dr. Sri Rama Murty for their interest in evaluating the progress of my research work. I am thankful to Mr. Debaditya Roy and Mr. Dinesh Singh for their valuable suggestions in thesis writing.

Last but not least, I sincerely thank my beloved parents for their constant support with their love and affection. I give my sincere thanks to my teachers, parents, grand parents, and colleagues for making me what I am today. Finally, I thank one and all who helped me directly or indirectly during my journey at IIT Hyderabad.

Shyju Wilson

ABSTRACT

Keywords: *Sparse Representation; Sparse Coding; Dictionary Learning; Matching Pursuit; Singular Value Decomposition; Information Bottleneck; Discriminative Dictionary; Jensen Shannon Divergence; Information Loss; Coherent Dictionary.*

The objective of this research work is to design compact and discriminative dictionaries for effective classification. The motivation stems from the fact that dictionaries inherently contain redundant dictionary atoms. This is because the aim of dictionary learning is reconstruction, not classification. In this thesis, we propose methods to obtain minimum number discriminative dictionary atoms for effective classification and also reduced computational time.

First, we propose a classification scheme where an example is assigned to a class based on the weight assigned to both maximum projection and minimum reconstruction error. Here, the input data is learned by K-SVD dictionary learning which alternates between sparse coding and dictionary update. For sparse coding, orthogonal matching pursuit (OMP) is used and for dictionary update, singular value decomposition is used. This way of classification though effective, still there is a scope to improve dictionary learning by removing redundant atoms because our goal is not reconstruction. In order to remove such redundant atoms, we propose two approaches based on information theory to obtain compact discriminative dictionaries. In the first approach, we remove redundant atoms from the dictionary while maintaining discriminative information. Specifically, we propose a constraint optimization problem which minimizes the mutual information between optimized dictionary and initial dictionary while maximizing mutual information between class labels and optimized dictionary. This helps to determine information loss between before and after the dictionary optimization. To compute information loss, we use *Jensen-Shannon diver-*

gence with adaptive weights to compare class distributions of each dictionary atom. The advantage of Jensen-Shannon divergence is its computational efficiency rather than calculating information loss from mutual information.

In the second approach, we propose a method to improve kernel K-SVD model as the kernelization of K-SVD results in better classification accuracy than its linear counter part. But the computation of kernel matrix incurs time and storage of the order $O(N^2)$ making it infeasible as the number of samples N grows. This can be solved by Nyström approximation of kernel matrix whose performance depends upon the underlying sampling strategy. So, we propose a sampling strategy based on information loss to improve Nyström approximation in linearization of kernel dictionary learning without affecting classification performance. Here, we find similar samples based on minimum information loss and merge them. This overall process results in kernelized features called virtual samples which can be directly applied to dictionary learning algorithms.

By leveraging the coherence of examples within a class, we propose another approach for obtaining compact and discriminative dictionary. It is observed that classes with high coherence can be represented with fewer dictionary atoms than classes with low coherence. Here, we divide the input data into coherent and non-coherent groups. The Coherent group consists of similar items whereas non-coherent has non-similar data items. For each class, we obtain dictionaries for coherent and non-coherent examples and treat them separately. Later coherent and non-coherent dictionaries are merged using Limited Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimization algorithm to form a single dictionary of particular class. We show that this obtained dictionaries achieve better classification than the dictionaries which are learned using all examples of a particular class. We demonstrate the efficacy of the proposed approaches on digit datasets and action datasets. The digit dataset helps to visualize discriminative dictionary atoms and make conclusions. We considered action dataset because it has structural sparsity in human motion and appearance.

In summary, this thesis proposes new methods for obtaining compact and discrim-

inative dictionaries based on information theory where redundant atoms are removed for better classification with low computational cost. We also propose a method for dictionary compaction based on coherent and non-coherent for better classification.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

OMP	- Orthogonal Matching Pursuit
SVD	- Singular Value Decomposition
MOD	- Method of Optimal Directions
JS	- Jensen Shannon
KL	- Kullback Leibler
DL	- Dictionary Learning
MP	- Matching Pursuit
BP	- Basis Pursuit
KSVD	- K Singular Value Decomposition
ODL	- Online Dictionary Learning
LASSO	- Least Absolute Shrinkage and Selection Operator
LARS	- Least Angle Regression
KNN	- K nearest neighbor
SRC	- Sparse Representation based Classification
FDDL	- Fisher Discriminative Dictionary Learning
LCKSVD	- Label Consistent KSVD
GP	- Gaussian Process
SVM	- Support Vector Machine
LKDL	- Linearized Kernel Dictionary Learning
MMI	- Maximization of Mutual Information

CHAPTER 1

INTRODUCTION TO DICTIONARY LEARNING

The evolving digital world witness explosion of large volume and complexity of data in this era of big data. The widespread availability of capturing devices and inexpensive data storage capability not only resulted in enormous amount of data, but also continuously escalate the growth of digital data around us. Social media sites, video, and image sharing sites are also important sources of vast data emergence in the digital world. The handling of the huge amount of data and extraction of information from the data are open issues now a days. The extraction of relevant information is more challenging because there is no optimum way to get relevant information from the large pool of data. So, finding discriminative and compact representation from codebook or dictionary has been widely addressed and relevant in this time [1–5]. The objective of this thesis is to propose methods to obtain discriminative information for classification tasks. More clearly, we build compact and discriminative dictionaries which are especially suitable for classification. In this work, the main motivation is that the dictionary obtained from standard learning algorithm inherently contains redundant dictionary atoms which are not necessarily useful for classification tasks.

1.1 DICTIONARY LEARNING

Sparse representation has been extensively applied in signal and image processing applications. It reconstructs the signals using a sparse set of fundamental units called atoms which form a structure referred to as *dictionary*. These atoms can be directly learned from the input samples rather than manually crafted mathematical functions such as wavelets [6], curvelets [7], contourlets [8], Bandelets [9] etc. The former approach of adaptive learning provides state of the art results compared to latter analytic methods. Using this dictionary, we reconstruct the signal using linear combination of atoms in the dictionary referred as *dictionary atoms*. Here, there are more number of dictionary atoms than the dimension of atom such that the given signal can have many different representations. This phenomenon is known as overcompleteness and dictionary is often called overcomplete dictionary. The process of determining coefficients for dictionary atoms in the signal reconstruction is called *sparse coding*. Each dictionary atom $\mathbf{d}_i \in R^m$ is denoted as column vector of m dimension. These atoms are concatenated to form dictionary as a matrix $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2 \ldots \mathbf{d}_K] \in R^{m \times K}$ and there are K atoms in the dictionary. Now we can approximate the signal $\mathbf{y}$ as a linear combination of atoms in the dictionary $\mathbf{D}$, ie., $\mathbf{y} \approx \mathbf{Dx}$, where $\mathbf{x} \in R^K$ is called sparse vector in which only few elements are non-zeros. It tells that the signal $\mathbf{y} \in R^m$ is reconstructed using sparsely determined dictionary atoms. The sparse vector $\mathbf{x}$ can be determined by any standard sparse coding like matching pursuit [10], basis pursuit [11] etc. In the same manner, we can find sparse vectors for all N input samples $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2 \ldots \mathbf{y}_N] \in R^{m \times N}$ and matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_N] \in R^{K \times N}$ contains corresponding sparse vectors of each sample in $\mathbf{Y}$.

Dictionary learning involves two steps, namely, sparse coding and dictionary update. After getting sparse matrix using any sparse coding algorithm, dictionary atoms are to be updated using input $\mathbf{Y}$ and corresponding sparse vectors in $\mathbf{X}$. In this learning process, it alternatively performs sparse coding and updation of dictionary in an iterative manner. Based on sparse coding and dictionary update, there exists different dictionary learning algorithms MOD, K-SVD etc. Finally, we will have dictionary D which represents the input data. Moreover, there is no universal dictionary to represent all kinds of signals in a sparse way. This calls upon the necessity to construct dictionary for each class or application.

1.2 ISSUES ADDRESSED IN THIS THESIS

For the classification purpose, discriminative dictionaries are to be created. So, the aim is to determine discriminative dictionary atoms which contain sufficient information with respect to particular class and are able to discriminate from other classes. In the context of discriminative dictionary learning, the dictionary is not necessary to be overcomplete because here the dictionary is meant for classification, not for signal reconstruction [12]. So, the dictionary learning contributes redundant dictionary atoms which can be eliminated while retaining discriminative dictionary atoms. These redundant atoms create additional computational burden in classification tasks. So, the main issue is to obtain compact and discriminative dictionaries for classification purpose. Here, we propose different ways to obtain compact discriminative dictionaries. Firstly, the input data is learned by K-SVD dictionary learning and then the learned dictionary is squeezed by an information theoretic approach often called *information bottleneck*. This is a constraint information theoretic problem in which mutual information between optimized dictionary and initial dictionary is minimized while maximizing mutual information between class labels and optimized dictionary. The redundant dictionary atoms can be effectively removed by considering change in information before and after the removal of dictionary atoms. The change in information or information loss can be calculated by computationally efficient distortion measure, *Jensen Shannon divergence*, using adaptive weights. Atoms which show least information loss are to be merged to remove the redundancy in the dictionary. The information bottleneck approach not only gives compact and discriminative dictionary, it also computationally efficient when compared with other state of the art methods.

Another issue of dictionary learning is its size ie., number of dictionary atoms in the learned dictionary. The ideal case is that minimum number of dictionary atoms which contain maximum discriminative information about class of input samples. In this work, we approximate the size of the dictionary by examining the amount of discriminative information while removing redundant atoms. In dictionary learning, constraint is given for number of dictionary atoms to be participated to reconstruct the given input signal. There are two types of constraints often used, namely, sparsity based and error based. Dynamically setting up the constraint based on the characteristics of each class is an another issue to be addressed. The dictionaries can be learned either using inputs from all classes together or separately for each of the classes. When we learn a single dictionary for all classes of inputs, there is an issue of determining the label of each learned dictionary atoms. This issue is addressed by examining distribution of dictionary atoms among the classes. The label of the dictionary atom is given based on the maximum number utilization of atom among classes.

To improve the recognition performance, recently many state of the art approaches [12–15] have been proposed kernelized dictionary learning. The main issue in the kernelization is the size of the kernel matrix which depends on the number of input samples. This large kernel matrix is computationally prohibitive when the number of input samples increases. To address this issue, the kernel matrix can be approximated using well-known Nyström method in which the subset of input data is used for the approximation. The selection of the subset of input data or sampling determines the quality of approximation. So, we propose an information loss based sampling for the Nyström approximation. In this, one dictionary atom $\mathbf{d}$ is removed from initial dictionary of particular class and sparse distributions of remaining dictionary atoms from the same class are found. Then the sparse distribution of $\mathbf{d}$ is to be compared with remaining dictionary atoms in order to find similar distribution. This proposed approach provides better sampling but it adds slight computational effort. After the Nyström approximation, we obtain kernelized feature vector called virtual samples which can be directly applied to any standard dictionary learning algorithms.

In another method, we propose to build compact and discriminative dictionary by exploiting underlying coherency among the samples. In this approach, the input data is divided into coherent and non-coherent groups. These two groups are learned and treated separately. Because of the similarity, coherent group can be learned into very few number of dictionary atoms while projection method is applied to include more independence among the dictionary atoms from non-coherent group. These two dictionaries from coherent and non-coherent group are concatenated and then updated to obtain single dictionary.

1.3 PRELIMINARIES

In this section, we discuss basic concepts required for the next chapters. We start with notations used in this thesis, and further state the definitions of entropy, mutual information and related concepts.

1.3.1 Notations

Here, we discuss notations used throughout this thesis. The bold small letters $(\mathbf{d}, \mathbf{y}, \ldots)$ represent vectors and bold capital letters $(\mathbf{D}, \mathbf{Y}, \ldots)$ represent matrices. For random variable notations, we use blackboard bold fonts $(\mathbb{D}, \mathbb{Y}, \ldots)$ and lowercase sans-serif letters $(\mathsf{d}, \mathsf{y}, \ldots)$ denote values taken by random variables. We denote probability mass function as $\mathsf{p}(\mathsf{d})$ and conditional distribution as $\mathsf{p}(\mathsf{y}|\mathsf{d})$ rather than $\mathsf{p}_{\mathcal{D}}(\mathsf{d})$ and $\mathsf{p}_{\mathcal{Y}|\mathcal{D}}(\mathsf{y}|\mathsf{d})$ for ease of use. The calligraphic notations $(\mathcal{D}, \mathcal{Y}, \ldots)$ for the spaces to which values of random variables belong. The unitalicized usual capital letters $(\mathrm{D}, \mathrm{Y}, \ldots)$ are used for set notations and sans-serif lowercase also used to denote values in set.

In this thesis, we use discrete random variables with a finite number possible values. That is, in our context, $(|\mathcal{D}|, |\mathcal{Y}|, \ldots)$ are all finite and $|\mathcal{D}|$ stands for cardinality of $\mathcal{D}$. Notation $\|.\|_p$ denotes l_p norm, commonly used values for p are 0, 1, and 2. The Frobenius norm for the matrix denoted as $\|.\|_F$. The notation $\sum_{\mathsf{x}}$ indicates the summation over all x values.

1.3.2 Linear algebra: Basics

Here, we discuss essential concepts of linear algebra used in this thesis. Many of these concepts are key to the problem formulations for research work carried out throughout this thesis.

Orthogonality

Let $\mathbf{x}$ and $\mathbf{y}$ be two vectors and perpendicular to each other called orthogonal, ie., $\mathbf{x} \perp \mathbf{y}$. Orthogonality holds if $\mathbf{x^T y} = \mathbf{0}$.

Projection

The vector $\mathbf{b}$ to be projected on the subspace spanned by independent vectors $\mathbf{a}_1, \mathbf{a}_2, \dots \mathbf{a}_n$ which form a matrix $A = [\mathbf{a}_1\, \mathbf{a}_2\, \dots\, \mathbf{a}_n]$. The $\mathbf{Ax}$ is projected vector on the subspace. So, $\mathbf{a}_1 \perp (\mathbf{b} - \mathbf{Ax})$, $\mathbf{a}_2 \perp (\mathbf{b} - \mathbf{Ax}), \dots \mathbf{a}_n \perp (\mathbf{b} - \mathbf{Ax})$. Now we can write

$$\mathbf{A}^T(\mathbf{b} - \mathbf{Ax}) = 0 \tag{1.1}$$

From the equation (1.1), we get

$$\mathbf{Ax} = \mathbf{A(A^T A)A^T b}$$

The matrix $\mathbf{A(A^T A)A^T}$ is called projection matrix.

Eigen decomposition

Let $\mathbf{A}$ be $n \times n$ matrix.

$$\mathbf{Ax} = \lambda\mathbf{x},$$

where λ is eigen value and $\mathbf{x}$ is eigen vector. Assume $\mathbf{A}$ has n independent eigen vectors $\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_n$. Then we can write

$$[\mathbf{Ax}_1 \quad \mathbf{Ax}_2 \dots \mathbf{Ax}_n] = [\lambda_1\mathbf{x}_1 \quad \lambda_2\mathbf{x}_2 \dots \lambda_n\mathbf{x}_n]$$
$$\mathbf{AX} = \mathbf{X\Lambda},$$

where $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2 \dots \mathbf{x}_n]$ and $\mathbf{\Lambda}$ is diagonal matrix whose diagonal contains eigen values. Then $\mathbf{A}$ can be decomposed as

$$\mathbf{A} = \mathbf{X\Lambda X^{-1}}$$

Positive definite symmetric matrix

For symmetric matrix, all eigen values are real and eigen vectors are perpendicular. In the case of positive definite symmetric matrix, all eigen values and pivots are positive. Suppose $\mathbf{A}$ is a positive definite matrix which holds $\mathbf{x^T A x} > 0$ for any vectors $\mathbf{x}$ except $\mathbf{x} = 0$.

Singular value decomposition (SVD)

For eigen decomposition, the matrix $\mathbf{A}$ to be $n \times n$ square matrix and it should have n independent eigen vectors, otherwise decomposing is not possible. The SVD is a way to decompose any rectangular matrix, ie., $\mathbf{A} \in R^{m \times n}$ and r is it's rank. As in the eigen decomposition, we can write

$$\mathbf{AV} = \mathbf{U\Sigma},$$

where $\mathbf{U} \in R^{m \times m}$, $\mathbf{V} \in R^{n \times n}$ be orthogonal matrices and $\mathbf{\Sigma} \in R^{m \times n}$ contains r singular values on diagonal and remaining values set to zero. Singular vectors $\mathbf{u}$'s and $\mathbf{v}$'s are obtained from

$$\mathbf{AA^T u}_i = \sigma_i^2 \mathbf{u}_i$$
$$\mathbf{A^T A v}_i = \sigma_i^2 \mathbf{v}_i$$

In the singular value decomposition, we can decompose $\mathbf{A}$ into a sum of r rank one matrices

$$\mathbf{A} = \mathbf{U\Sigma V}^T = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \ldots + \sigma_r \mathbf{u}_r \mathbf{v}_r^T$$

1.3.3 Entropy

Entropy is the measure of uncertainty contained in a random variable. The entropy is denoted as $H(\mathbb{X})$ or $H[\mathsf{p}(\mathsf{x})]$ where random variable $\mathbb{X}$ has distribution according to probability mass function $\mathsf{p}(\mathsf{x})$, ie. $\mathsf{p}(\mathbb{X} = \mathsf{x})$, $\mathsf{x} \in \mathcal{X}$. The entropy only depends on

$\mathsf{p(x)}$, not on the actual values of x. Then the entropy $H(\mathbb{X})$ of the discrete random variable can be defined as

$$H(\mathbb{X}) = -\sum_{\mathsf{x}\in\mathcal{X}} \mathsf{p(x)} \log \mathsf{p(x)}.$$

The $\log$ is to the base 2 and entropy is expressed in bits. We use the convention that $0 \log 0 = 0$ since $\mathsf{x} \log \mathsf{x} \to 0$ as $\mathsf{x} \to 0$. Another important fact is that entropy is always non negative. When $\mathsf{p} = \frac{1}{|\mathcal{X}|}$, then the entropy is maximum and it is monotonically increasing function of $|\mathcal{X}|$. Suppose $|\mathcal{X}| = 2$

$$H(\mathbb{X}) = -\mathsf{p} \log \mathsf{p} - (1 - \mathsf{p}) \log (1 - \mathsf{p}).$$

We can see $H(\mathbb{X}) = 1$ when $\mathsf{p} = \frac{1}{2}$ and it gives concave function as shown in figure 1.1. The $H(\mathbb{X}) = 0$ when $\mathsf{p} = 0$ or 1 which means the variable does not have randomness, so there is no place for uncertainty. Moreover, entropy is also a lower bound of the average number of bits needed to represent a random variable. Now we extend the definition of entropy to more than one random variable.

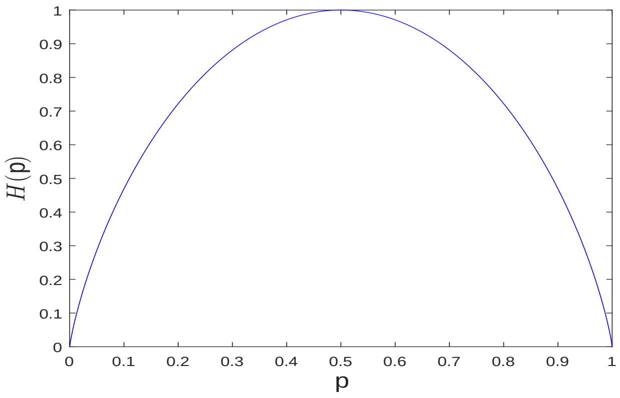


Fig. 1.1: $H(\mathsf{p})$ *vs.* p

Joint entropy

Suppose $\mathbb{X}$ and $\mathbb{Y}$ are two random variables and joint probability distribution is denoted as $\mathsf{p}(\mathsf{x},\mathsf{y})$. Then we can define joint entropy as

$$H(\mathbb{X},\mathbb{Y}) = -\sum_{\mathsf{x}\in\mathcal{X}}\sum_{\mathsf{y}\in\mathcal{Y}} \mathsf{p}(\mathsf{x},\mathsf{y}) \log \mathsf{p}(\mathsf{x},\mathsf{y})$$

where joint entropy tells the uncertainty over the random variables $\mathbb{X}$ and $\mathbb{Y}$.

Conditional entropy

Let $\mathbb{X}$ and $\mathbb{Y}$ are two random variables, then the conditional entropy can be defined as

$$\begin{aligned} H(\mathbb{Y}|\mathbb{X}) &= \sum_{\mathsf{x}\in\mathcal{X}} \mathsf{p}(\mathsf{x})\ \mathsf{H}(\mathbb{Y}|\mathbb{X}=\mathsf{x}) \\ &= -\sum_{\mathsf{x}\in\mathcal{X}}\sum_{\mathsf{y}\in\mathcal{Y}} \mathsf{p}(\mathsf{x},\mathsf{y}) \log \mathsf{p}(\mathsf{y}|\mathsf{x}), \end{aligned} \tag{1.2}$$

where $H(\mathbb{Y}|\mathbb{X})$ indicates the uncertainty remaining over $\mathbb{Y}$ after knowing value of $\mathbb{X}$. By using equation (1.2), we can rewrite $H(\mathbb{X},\mathbb{Y})$ as

$$H(\mathbb{X},\mathbb{Y}) = H(\mathbb{X}) + H(\mathbb{Y}|\mathbb{X}). \tag{1.3}$$

1.3.4 Mutual Information and related concepts

Mutual information is the amount of information that one random variable contains about another. In other words, it is the reduction in uncertainty of one random variable by knowing other one. Suppose $\mathbb{X}$ and $\mathbb{Y}$ are two random variables having joint probability mass distributions $\mathsf{p}(\mathsf{x},\mathsf{y})$ while $\mathsf{p}(\mathsf{x})$ and $\mathsf{p}(\mathsf{y})$ are marginal probability mass functions. The mutual information among $\mathbb{X}$ and $\mathbb{Y}$ can be defined as

$$I(\mathbb{X};\mathbb{Y}) = -\sum_{\mathsf{x}}\sum_{\mathsf{y}} \mathsf{p}(\mathsf{x},\mathsf{y}) \log \frac{\mathsf{p}(\mathsf{x},\mathsf{y})}{\mathsf{p}(\mathsf{x})\mathsf{p}(\mathsf{y})}$$

The mutual information is also written in terms of entropy

$$I(\mathbb{X};\mathbb{Y}) = H(\mathbb{X}) - H(\mathbb{X}|\mathbb{Y})$$

From equation (1.3), we can write

$$I(\mathbb{X};\mathbb{Y}) = H(\mathbb{X}) + H(\mathbb{Y}) - H(\mathbb{X},\mathbb{Y})$$

The figure 1.2 shows the relation between mutual information and entropy.

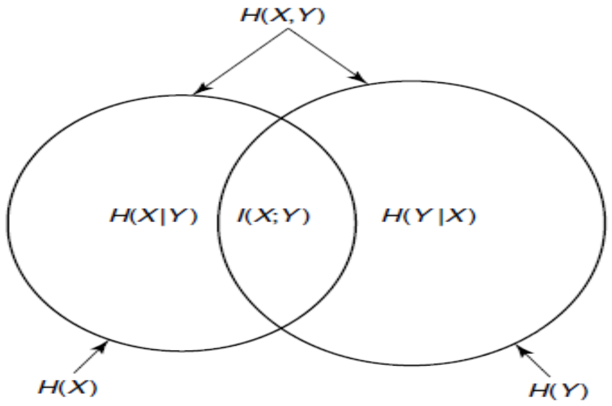


Fig. 1.2: The relationship between entropy and mutual information

We discuss two important distortion measures, namely, Kullback-Leibler divergence or relative entropy and Jensen-Shannon divergence. It measures the distance between two probability distributions.

Relative entropy

The relative entropy of two probability mass functions $\mathsf{p(x)}$ and $\mathsf{q(x)}$ can be defined as

$$\mathsf{D(p||q)} = \sum_{\mathsf{x}} \mathsf{p(x)} \log \frac{\mathsf{p(x)}}{\mathsf{q(x)}}.$$

We follow the convention that $\mathsf{0} \log \frac{\mathsf{0}}{\mathsf{0}} = \mathsf{0}$, $\mathsf{0} \log \frac{\mathsf{0}}{\mathsf{q}} = \mathsf{0}$ and $\mathsf{0} \log \frac{\mathsf{p}}{\mathsf{0}} = \infty$. The relative entropy is non-negative and becomes zero whenever $\mathsf{p} = \mathsf{q}$, ie., $\mathsf{D(p||q)} \geq \mathsf{0}$. Since it is not a symmetric, it cannot be considered as true distance between distributions. However, it is often considered as the distance between two distributions. Then the mutual information can be rewritten in the form of relative entropy, ie., $I(\mathbb{X};\mathbb{Y}) = \mathsf{D(p(x,y)||p(x)p(y))}$. Since $\mathsf{D(p||q)} \geq \mathsf{0}$, the quantity of mutual information is also a non-negative, ie., $I(\mathbb{X};\mathbb{Y}) \geq 0$.

Jensen-Shannon divergence

Jensen's inequality and *Shannon's entropy* derive the distortion measure Jensen-Shannon divergence (JS divergence). For J directed divergence [16] and its symmetric measure I divergence, both distributions should be *absolutely continuous* with respect to each other. But Jensen-Shannon divergence does not have this kind of issue. Here, prior probabilities (weights) can be assigned to different probability distributions which ultimately improves decision problems. For the Bayes' probability of misclassification error [17], the distortion measure provides both lower and upper bound. Another important feature of JS divergence is that it can be generalized for more than two distributions. Let $\mathsf{p}_1, \mathsf{p}_2 \ldots, \mathsf{p}_\mathsf{n}$ be n probability distributions with weights $\pi_1, \pi_2, \ldots, \pi_\mathsf{n}$, respectively, and $\sum_\mathsf{i} \pi_\mathsf{i} = 1$. The generalized Jensen-Shannon can be defined as,

$$\mathsf{JS}_\pi(\mathsf{p}_1, \mathsf{p}_2, \ldots, \mathsf{p}_\mathsf{n}) = H\Big(\sum_\mathsf{i} \pi_\mathsf{i} \mathsf{p}_\mathsf{i}\Big) - \sum_\mathsf{i} \pi_\mathsf{i} H(\mathsf{p}_\mathsf{i}).$$

where H denotes entropy.

1.4 ORGANIZATION OF THE THESIS

An overview of the existing approaches to discriminative dictionary learning is discussed in Chapter 2. The Chapter 3 explains how the sparsity based dictionary learning can be applied for classification problems based on minimum reconstruction error as discriminative measure. The Chapter 4 contains the proposed method to build compact and discriminative dictionary especially for classification using information bottleneck approach. The linearization of kernel dictionary learning using Nyström approximation with information loss based sampling is proposed in Chapter 5. In Chapter 6, we propose another method to obtain discriminative dictionary in which the input data is divided into coherent and non-coherent and treated them separately. Chapter 7 summarizes the entire research work carried out as part of this thesis, glimpses the important contributions of the research and gives directions towards future works.

CHAPTER 2

OVERVIEW OF DICTIONARY LEARNING

This chapter reviews evolution and existing approaches of dictionary learning (DL) for signal representation and discusses notable works to build discriminative dictionaries. We describe the evolution of dictionary design until the recent time in section 2.1. The section 2.2 details about different analytic dictionaries and section 2.3 discusses existing approaches to train the dictionaries. In the section 2.4, we identify the quest for discriminative dictionaries for classification tasks. In the section 2.5, we discuss the issues in the existing discriminative dictionary learning approaches and finally section 2.6 summarizes the overall review.

2.1 EVOLUTION OF DICTIONARY DESIGN

Signal processing techniques demand useful representations which contain the important nature of the signal. This representation should (1) possess relevant features for recognition; (2) efficiently separate noise from signal for denoising; and (3) capture useful part of the signal with only a few coefficients for compression. Signal representation involves the selection of a dictionary, which contains fundamental signals or atoms, used for the decomposition of a signal. When the dictionary becomes a basis, then each of the signal can be represented uniquely using the linear combination of atoms in the dictionary. The simplest one is orthogonal dictionary in which representation coefficients are computed using inner product of the atoms and signal. In the case of non-orthogonal, the inner product of the the dictionary inverse and signal determine the coefficients, also called bi-orthogonal dictionary. These bi-orthogonal and orthogonal dictionaries were popular due to its simplicity in mathematical formulations, but lack expressiveness. This is the reason behind the introduction of overcomplete dictionaries in which it has more number of atoms than the dimension of the signal, which intended to include more diverse spectrum of signal characteristics. In this section, we describe the evolution of dictionary design methodologies especially from analytic to adaptive learning.

2.1.1 Linear Model

In 1960's, Fourier transform [18] played an important role to describe a signal with respect to its whole frequency content. The signal is approximated using the projection of basis onto the K atoms which has low frequency components, it has a strong noise-reducing and smoothing effect. So, the Fourier basis can be efficiently used to describe uniformly smooth signals, but difficult to represent discontinuities. The discrete cosine transform (DCT) [19] gives more efficient representation which results in continuous boundary. The advantage of discrete cosine transform is that it produces non-complex coefficients which are preferably considered in practical applications. The statistical tool Karhunen-Loeve transform (KLT) [20] is another linear transform which can be used to represent the signals obtained from a particular kind of known distributions. Atoms belong to the KLT are taken from the eigenvalue decomposition of the data covariance matrix such that first K eigenvectors are selected, which fits the subspace spanned over the low dimension to the data while minimizing error approximation using l_2 norm. This adaptation process has good representational efficiency compared with the Fourier transform, but its transformation is complex. In modern dictionary design, we will see that the trade-off between *adaptivity* and *efficiency* which plays an important role in it.

2.1.2 Non-Linear Model

In 1980's and 1990's, sparsity plays a major role in the field of signal analysis and recovery and origin of this idea goes back to classical physics and information theory.

During this time, the researchers actively worked for more efficient transforms and sparse representations for signal processing tasks . The enforcement of sparsity led the transformation of linear model to non-linear model which is having more flexible formulation. In this case of non-linear, each signal is approximated using various set of dictionary atoms which is preferably sparse set and this pave the way for efficient transforms. Better localization in transforms helps to achieve sparsity. The concentrated support of atoms give much flexible representations based on the limit the effects of irregularities and local signal characteristics. The short time fourier transform (STFT) [21] was one of the first structures used this, naturally STFT becomes the extension of the Fourier transform. To obtain space-frequency or time-frequency characteristcs of the signal, the application of the Fourier transform is considered locally on portions of the signal which might be possibly overlapping. This is also known as Gabor transform [22,23]. Daubechies *et al.* [24,25] contribute mathematical founadations of Gabor transform and discrete versions of this transform are given by Wexler *et al.* [26] and Qian *et al.* [27]. The development of complex Gabor structures for higher dimensions included directionality which is obtained by changing the orientation in the sinusoidal signals. Daugman [28, 29] used this structure to discover important phenomena that the simple cell receptive area of the visual cortex has the pattern like oriented Gabor structure. These developments led to the intensive use of the transform in the areas of applications in image processing [30, 31]. Now, Gabor transforms are used in directional filters for analysis and detection tasks. The multi resolution [32] is another advancement in which natural signals especially images showed relevant information about structures using different scales and analysis of the signal could be done in an efficient manner.

Another breakthrough came in mid 1980's called wavelet analysis [20, 33] which proposes expansion of signal using set of dilated and translated versions of the single fundamental function . Mallat *et al.* [34–36] described a pair of localized functions, namely, *scaling function* and *mother wavelet* from which multi-scale wavelet basis was constructed. The low frequency signals are contained in scaling function whereas high frequency signal contents in the mother wavelet, and signals are described using its different translations and scales. The non-linear approximation using wavelet basis has been used by piecewise smooth one dimensional signals having few discontinuities, which was shown to be optimal [37] but wavelet transform loses its optimality in higher dimensions. To overcome the limit of approximation in orthogonal bases, transform atoms can be adapted to the signal content. Coifman *et al.* [38] proposed wavelet packet transform and added adaptivity which gives finer tuning to certain kind of signal characteristics. However, the multi-dimensional wavelet packet transform could not give a notable improvement over the wavelets for images. For the dictionary property of the invariance under certain geometric deformations, Simoncelli *et al.* [39] suggest overcompleteness while abandoning orthogonality. The stationary wavelet transform is an undecimated transform which substantially improves recovery of signals when comparing with orthogonal wavelets [40, 41].

2.1.3 Dictionaries

The *dictionaries* for sparse signal representations replaced *transforms* by the second half of 1990's. Mallat and Zhang [10] sparsely expand the signal using few elementary functions in overcomplete dictionary of functions. This is popularly known as *Matching Pursuit* (MP). Later, Chen *et al.* [11] published similar kind of work called *Basis Pursuit* (BP). These two pioneer works signalled the beginning of new era in modern signal processing [42]. In this, the main intuition is that a signal can have many description in the domain of representation, and choose the best one which suit for particular task.

The dictionaries of analytic formulation model a signal of interest by simple set of mathematical functions and use this model to design efficient representations. The wavelet dictionary contains piecewise smooth functions and point singularities whereas the Fourier dictionary includes smooth functions . These kind of dictionaries have the advantage of fast and efficient implementation because the computation does not include any multiplication with the dictionary matrix, but not good to capture complexity of the natural phenomena of signals. This difficulty can be solved by example-based learning which led to trained dictionaries. The intuition behind this learning method is that it can capture complexity of natural phenomena of signals from the data directly rather than using a mathematical description. The details of analytic and trained dictionaries are discussed in the following sections.

2.2 ANALYTIC DICTIONARIES

The formulation of analytic dictionaries become generally as tight framework, ie., $\mathbf{DD^T y} = \mathbf{y}$ for all $\mathbf{y}$, in this case dictionary's transpose has been used to get the representation of the signal over the dictionary. So, analysis operator $\mathbf{D^T}$ can be easily analyzed as compared to a synthesis model in which sparsity constraint to be derived. This method provides an efficient and simple procedure to attain sparse representations using atoms in the dictionary. If we look at from the angle of synthesis point of view, this analytic process is sub-optimal.

2.2.1 Curvelets

In 1999, Candes and Donoho [43] introduce curvelet transform, and later it was refined into its current form in early 2000's [44]. At an optimal rate, it represented two dimensional piecewise smooth functions with smooth curve discontinuities and the elongated elliptical region supports curvelet atoms which are oscillatory along its width, smooth along its length. But these curvelet atoms are become flattened ellipsoids which oscillate along shorter directions and smooth along the other directions [44, 45] in higher dimensions.

2.2.2 Contourlets

Despite the curvelet transform provides a solid continuous construction, its discretization found to be difficult. The existing discretizations have relatively high redundancies, so not suitable for tasks like compression. To overcome these limitations, Do *et al.* [8, 46] proposed an alternative to the two dimensional curvelet transform, which is called contourlet transform. However, the major setback in the construction of contourlet is that the basis images are not localized in its frequency domain. Later, this transform was improved by Lu *et al.* [47] by introducing a new multiscale decomposition in its frequency domain. The contourlet transform has many features of the curvelet transform, namely, parabolic scaling, localization, and orientation, but the difference is that contourlets have been defined in the discrete domain which advocated to construct discrete signals in an efficient manner. As compared to improved curvelets [44], the original contourlet transform exhibits lower redundancy, so it can be used for the application like image compression. Though this transform is apt for the image compression, its enormous sub-sampling produces artifacts in signal reconstruction. To counter this issue, translation invariant [48] and non sub-sampled [49] version of the transform is considered, but this option raises complexity and redundancy.

2.2.3 Bandelets

Le Pennec and Mallat [9] proposed the bandelet transform which was later modified by Peyre *et al.* [50]. Unlike the non-adaptive contourlet and curvelet transforms, the bandelet transform is pioneer step in the area of adaptive signal transforms. The bandelet construction exploits the geometric regularities exist in the images, especially directional characteristics and edges, which helps to fit the specific set of optimized atoms to an image. With respect to dictionaries, the bandelet transform chooses group of atoms from a nearly infinite set, and the discretization limits the size of this set. But, in the wavelet packet transform, full set of atoms is not much larger than signal's dimension.

The complex wavelet transform [51], shearlet transform [52], directionlet transform [53], grouplet [54] transform are other important analytic dictionaries.

2.3 DICTIONARY TRAINING

The recent approaches for dictionary training has been deeply motivated from the recent developments in signal representation using sparse based approaches. The l_0 and l_1 sparsity measures are used in most recent training methods, which result to simple and efficient formulation. These measures can be applied in modern sparse coding techniques [10, 55]. The major contribution in the field of dictionary learning was given by Olshausen *et al.* [56]. The authors used small patches of images as dictionary atoms and train the dictionary for sparse representation. The obtained trained atoms were similar to the mammalian simple cell receptive fields, previously Gabor filters weakly explained this receptive fields.

2.3.1 Method of Optimal Directions

Engan *et al.* [57] introduced one of the first methods to implement modern sparse dictionary learning known as Method of Optimal Directions (MOD) proposed in 1999. This kind of implementation paved new way for modern dictionary learning. The given set of input examples $\mathbf{Y} = [\mathbf{y_1}\,\mathbf{y_2}\ldots\mathbf{y_N}] \in \mathbf{R^{m\times N}}$, the goal of this approach is to find sparse matrix $\mathbf{X} = [\mathbf{x_1}\,\mathbf{x_2},\ldots\mathbf{x_N}] \in R^{K\times N}$ and dictionary $\mathbf{D} = [\mathbf{d_1}\,\mathbf{d_2}\ldots\mathbf{d_K}] \in R^{m\times K}$ by minimizing representation error

$$\underset{\mathbf{D},\mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{DX}\|_{\mathbf{F}}^{\mathbf{2}} \quad \text{subject to} \quad \forall \mathbf{i}\ \|\mathbf{x_i}\|_{\mathbf{0}} \leq \mathbf{T}, \tag{2.1}$$

where T and $\mathbf{x}$ are sparsity constraint and sparse vector, respectively. Notations $\|.\|_0$ and $\|.\|_F$ denote l_0 norm and Frobenius norm, respectively. This resulting optimization problem is highly non-convex, so optimization function finds its local minimum at best. The MOD performs sparse coding and dictionary update alternatively as followed in the similar training methods. In sparse coding stage, sparse coefficients over dictionary are determined for each signal separately using any standard sparse coding algorithm. And this dictionary can be updated in direct way, $\mathbf{D} = \mathbf{Y}\mathbf{X}^{\mathbf{T}}(\mathbf{X}\mathbf{X}^{\mathbf{T}})^{-\mathbf{1}}$ by solving (2.1). The MOD needs only few iterations to converge and reaches at local minima. This dictionary update involves matrix inverse which claims relatively much computational complexity. Many subsequent works are concentrated on reducing this complexity, which led other state of the art approaches to train the dictionary efficiently.

2.3.2 Online dictionary learning

Online dictionary learning (ODL) [58] is highly used in the area of image and video processing, which is able to handle large datasets and computationally very effective. The ODL has two important steps: one is sparse coding in which l_1 norm based regularization used as follows

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_F^2 + \lambda\|\mathbf{x}\|_1, \tag{2.2}$$

where λ denotes regularization parameter and the constraint l_1 norm is applied on the sparse vector $\mathbf{x} \in R^K$. To obtain sparse solution, ODL uses least-angle regression (LARS) [59] which efficiently implements the least absolute shrinkage and selection operator (LASSO) [60]. The lasso is l_1 regularized selection procedure known as *basis pursuit* [11,55]. In the second step of ODL, the dictionary atoms are updated using the obtained sparse vectors $\mathbf{x}_i$'s and their corresponding input $\mathbf{y}_i$'s. The block coordinate descent is used to update each dictionary atom and new dictionary is obtained by minimizing the optimization function

$$\underset{\mathbf{D} \in C}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_F^2 + \lambda\|\mathbf{x}_i\|_1, \tag{2.3}$$

where C denotes convex set of matrices having the following constraint

$$C \doteq \{\mathbf{D} \in R^{m \times K} \text{ s.t. } \quad \forall_i = 1, \ldots, K \quad \mathbf{d}_i^T \mathbf{d}_i \leq 1\}.$$

2.3.3 The K-SVD algorithm

Another efficient dictionary training for sparse signal representation was proposed by Aharon *et al.* [61] in 2005, which is known as K-SVD dictionary learning. The K-SVD uses sparse coding algorithm orthogonal matching pursuit (OMP) in which l_0 norm is used for the constraint. The OMP [62] is a modification of *matching pursuit* given by Mallat and Zhang [10]. The K-SVD is an improved version of MOD [57] in which pseudo inverse is used to update the dictionary, whereas K-SVD uses singular value decomposition (SVD) for the updation. As shown in equation 2.1, K-SVD also performs sparse coding using OMP and dictionary update using SVD alternatively. In this process of updation, each of the dictionary atom is updated sequentially. There are K dictionary atoms, so it has to run K times. For instance, the k^{th} dictionary atom $\mathbf{d}_k$ to be updated, error matrix $\mathbf{E}_k$ is obtained by removing $\mathbf{d}_k$ and corresponding sparse coefficients from error equation $\mathbf{Y} - \mathbf{D}\mathbf{X}$, i.e. $\mathbf{E}_k = \mathbf{Y} - \sum_{j \neq k} \mathbf{d}_j \mathbf{x}^j$, where $\mathbf{x}^j$ is the j^{th} row of sparse matrix $\mathbf{X}$, that corresponds to dictionary atom $\mathbf{d}_j$. Now the optimization function can be rewritten as

$$\begin{aligned} \left\| \mathbf{Y} - \mathbf{D}\mathbf{X} \right\|_F^2 &= \left\| \mathbf{Y} - \sum_{j \neq k} \mathbf{d}_j \mathbf{x}^j - \mathbf{d}_k \mathbf{x}^k \right\| \\ &= \left\| \mathbf{E}_k - \mathbf{d}_k \mathbf{x}^k \right\|_F^2 . \end{aligned} \tag{2.4}$$

To maintain sparsity, the input samples which are not used by atom $\mathbf{d}_k$ can be removed. This can be achieved by removing zero coefficients from $\mathbf{x}^k$ and denoted as $\mathbf{x}_R^k$, then corresponding columns from $\mathbf{E}_k$ to be removed and denoted as $\mathbf{E}_k^R$. The equation (2.4) is rewritten as $\|\mathbf{E}_k^R - \mathbf{d}_k \mathbf{x}_R^k\|_F^2$ and matrix $\mathbf{E}_k^R$ to be decomposed by SVD to update dictionary atom $\mathbf{d}_k$.

2.4 DISCRIMINATIVE DICTIONARY LEARNING

Discriminative dictionary learning finds dictionary especially for classification tasks because standard dictionary learning aims to represent training samples, not suitable for classification [63]. In this, the important thing is the discriminative ability of the dictionary to classify the input example from other classes. For face recognition, a sparse representation based classification (SRC) was proposed by Wright *et al.* [64] and reconstruction error is used as discriminative measure which results better performance. This is a naive way of doing classification using dictionary and authors never try to incorporate any discriminative component in dictionary learning for classification. Mairal *et al.* [65] added a discriminative reconstruction constraint in K-SVD dictionary learning to add discriminative ability among dictionaries, and these learned dictionaries are used for texture segmentation and scene analysis. However, this method does not utilize the discriminative ability of sparse coding coefficients. This is considered in the improved version of [65] in which Mairal *et al.* [66] proposed a discriminative dictionary learning by training a classifier for the sparse coding coefficients, and applied this method for digit recognition and texture classification. In [67], more discriminative terms are added for tuning the dictionary learning into specific task like semi-supervised learning. In this, unlabelled data are exploited by sparse representation and effectively applied task like classification.

Pham *et al.* [68] trained linear classifier from dictionary and then the dictionary is updated from the learned classifier. In this, it alternates until convergence which results in better discriminative sparse representation for face recognition. As an extension to [68], Zhang *et al.* [69] proposed discriminative KSVD (DKSVD) for face recognition. All these works in [66, 68, 69] learned a dictionary in which all classes share dictionary atoms and also learned a classifier of coefficients for the classification purpose. However, the shared dictionary may loose the correspondence between the class labels and the dictionary atoms, and whenever the number of classes and size of training samples increase, the computational complexity of dictionary training becomes high. Then for each class, Yang *et al.* [70] learned separate dictionary and obtained impressive results for face recognition. Ramirez *et al.* [71] suggested specific term for incoherence which is intended to keep the dictionaries of different classes as independent as possible. Lobel *et al.* [1] used dictionary of linear classifiers to encode mid level representations from different regions of an image. These classifiers are applied to max pooling strategy and feature descriptor to make total energy of an image as linear combination of max functions to obtain discriminative and compact visual words. Liu *et al.* [4] introduced probabilistic framework for merging criteria to produce well representative codebook. As we have seen, generally, reconstruction error corresponds to classes has been used as the discriminative information for classification. Here, we discuss some of the important works in discriminative dictionary learning.

2.4.1 Fisher discriminative dictionary learning (FDDL)

Yang *et al.* [63] proposes a new way to obtain discriminative dictionary for the classification purpose by incorporating modification to the reconstruction error function posed in (2.1). This new discriminative learning framework includes the Fisher discrimination criterion [72] to learn a structured dictionary in which dictionary atoms are associated with class labels, so that for classification, the reconstruction error corresponds to each class has been used. The Fisher discrimination criterion is applied on the sparse coding coefficients apart from imposing class specific constraints which ultimately produce discriminative sub-dictionaries for classification. In this method, the dictionary is split into n disjoint sets which indicate different classes.

Let $\mathbf{Y} = [\mathbf{Y}_1 \ \mathbf{Y}_2 \ldots \mathbf{Y}_n] \in R^{m \times N}$ be the N input samples from all classes and $\mathbf{Y}_i = [\mathbf{y}_1 \ \mathbf{y}_2 \ldots \mathbf{y}_{n_i}] \in R^{m \times n_i}$ are the samples belong to class i. The dictionary $\mathbf{D} = [\mathbf{D}_1 \ \mathbf{D}_2 \ldots \mathbf{D}_n] \in R^{m \times K}$ and sparse matrix $\mathbf{X} = [\mathbf{X}_1 \ \mathbf{X}_2 \ldots \mathbf{X}_n] \in R^{K \times N}$ are obtained from the given input $\mathbf{Y}$. The set $\mathbf{X}_i \in R^{K \times n_i}$ can be further decomposed as $\mathbf{X}_i = [(\mathbf{X}_i^1)^T \ldots (\mathbf{X}_i^j)^T \ldots (\mathbf{X}_i^L)^T]$, where $\mathbf{X}_i^j \in R^{k_j \times n_i}$ are the coefficients obtained using samples $\mathbf{Y}_i \in R^{m \times n_i}$ over the dictionary atoms in $\mathbf{D}_j \in R^{m \times k_j}$. Now the objective function is formulated as similar to [69] for discriminative dictionary learning. The objective function in FDDL consists of two parts. In the first part, authors try to improve the reconstruction error function posed in (2.1) and in the second part, Fisher discriminative criterion is added for further improvement of discriminability in the dictionary. The first part is formulated as

$$r(\mathbf{Y}_i, \mathbf{D}, \mathbf{X}_i) = \|\mathbf{Y}_i - \mathbf{D}\mathbf{X}_i\Big\|_F^2 + \|\mathbf{Y}_i - \mathbf{D}_i\mathbf{X}_i^i\Big\|_F^2 + \sum_{\substack{j=1\\j\neq i}}^{L} \|\mathbf{D}_j\mathbf{X}_i^j\Big\|_F^2. \tag{2.5}$$

The first term for the representation of samples in the i^{th} class based on all dictionary atoms whereas second term utilizes only dictionary atoms belong to the i^{th} class for the representation. The third term enforces self-reliance on i^{th} class and reduces the relation with other classes. Fisher Discriminant Criterion has been used in the second part of optimization formulation. Two scatter functions are used for the representation, one is for *within* class $S_W(\mathbf{X})$ and another is for *between* class $S_B(\mathbf{X})$.

$$\begin{aligned} S_W(\mathbf{X}) &= \sum_{i=1}^{L} \sum_{\mathbf{x}_k \in X_i} (\mathbf{x}_k - \mu_i)(\mathbf{x}_k - \mu_i)^T, \\ S_B(\mathbf{X}) &= \sum_{i=1}^{L} n_i(\mu_i - \mu)(\mu_i - \mu)^T, \end{aligned} \tag{2.6}$$

where $\mu, \mu_i \in R^{K\times 1}$ denote mean vectors of sparse vectors in $\mathbf{X}$ and $\mathbf{X}_i$, respectively. Here, we minimize the function $S_W(\mathbf{X})$ while maximizing $S_B(\mathbf{X})$, then combine equation (2.5) and (2.6) to obtain final objective function

$$\underset{\mathbf{D},\mathbf{X}}{\operatorname{argmin}} \sum_{i=1}^{L} r(\mathbf{Y}_i, \mathbf{D}, \mathbf{X}_i) + \lambda_1\|\mathbf{X}\|_1 + \lambda_2[tr(S_W(\mathbf{X}) - S_B(\mathbf{X})) + \eta\|\mathbf{X}\|_F^2].$$

The term $\|\mathbf{X}\|_F^2$ serves as a regularization. Based on this work, Cai *et al.* [73] introduce a discriminative dictionary learning model based on support vector. In [74], authors used Fisher discriminative dictionary learning to map data from various modalities to common subspace in which inherent relationship between different modalities become evident.

2.4.2 Label-Consistent KSVD (LC-KSVD)

The other discriminative dictionary learning method has been proposed in [75, 76]. In this classification parameters are passes along with dictionary learning parameters ie., all the parameters are combined to form one objective function are learned by standard K-SVD dictionary learning algorithm. Then the additional terms are added to the standard optimization function as

$$\underset{\mathbf{D},\mathbf{T},\boldsymbol{\Theta},\mathbf{X}}{\text{argmin}}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\Big\|_F^2+\alpha\|\mathbf{Q}-\mathbf{T}\mathbf{X}\Big\|_F^2+\|\mathbf{H}-\boldsymbol{\Theta}\mathbf{X}\Big\|_F^2 \quad \text{s.t.} \quad \forall i \ \ \|\mathbf{x_i}\|_\mathbf{0}\leq \mathbf{q}. \tag{2.7}$$

The second term encourages the sparse coefficients to be discriminative. The matrix $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots \mathbf{q}_N] \in R^{K\times N}$ denotes sparse matrix for discrimination in which the coefficient $\mathbf{q}_{i,j}$ is 1 if the class of the dictionary atom $\mathbf{d}_i$ matches with input signal $\mathbf{y}_j$ and 0 if they do not match. This term encourages similar sparse code for the input samples belong to the same class than sparse codes from other classes. One more term is added for classification error in which $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots \mathbf{h}_N] \in R^{L\times N}$ denotes label matrix that corresponds to all input samples where $\mathbf{h}_{i,j} = 1$ if the input sample $\mathbf{y}_i$ belong to the j^{th} class and 0 otherwise. Now the optimization function becomes

$$\underset{\mathbf{D}_{new},\mathbf{X}}{\text{argmin}}\|\mathbf{Y}_{new}-\mathbf{D}_{new}\mathbf{X}\Big\|_F^2 \quad \text{subject to} \quad \forall i \ \ \|\mathbf{x_i}\|_\mathbf{0}\leq \mathbf{q}, \tag{2.8}$$

where $\mathbf{Y}_{new} = (\mathbf{Y}^T, \sqrt{\alpha}\mathbf{Q}^T, \sqrt{\beta}\mathbf{H}^T)^T \in R^{(m+K+L)\times N}$ and $\mathbf{D}_{new} = (\mathbf{D}^T, \sqrt{\alpha}\mathbf{T}^T, \sqrt{\beta}\boldsymbol{\Theta}^T)^T \in R^{(m+K+L)\times K}$. Now the standard K-SVD algorithm has been used to solve the optimization function obtained in (2.8).

2.4.3 Information theoretic approaches for discriminative dictionary

Many machine learning applications have been used the mutual information as a similarity measure [3,77]. In [78] [79], Krause *et al.* worked on optimal placement of sensors to measure temperature based on Gaussian process (GP) using maximum mutual information which ultimately reduces the communication costs. Based on this work in [78], Qiu *et al.* [3] learnt input data using K-SVD dictionary learning and then choose atoms by maximizing mutual information between chosen and non-chosen atoms. To ensure enough representation of all classes in the learned dictionary, they also maximize mutual information among classes. But *Gaussian Process* (GP) model is used for sparse representation, so the matrix inverse is to be computed which claims more computational time. In [2], authors maximize mutual information between chosen and non-chosen atoms, between class labels and sparse codes, between selected atoms and input signals, and then gradient ascent algorithm is used to update the dictionary to obtain discriminative dictionary.

Liu and Shah [80] extract 3D interest points refers to video words and optimize these video words by maximizing mutual information to learn human actions. Lee *et al.* [77] determine similaity between two activity vectors which are obtained from different cameras based on maximization of mutual information. In [81], codebooks are learned by minimizing the loss of information for image classification and segmentation. Information theoretic approaches are effective measure to calculate the amount of information retains after learning dictionary from input data. In the deep neural network, Tishby *et al.* [82] systematically measure loss of information while learning through each layer.

2.5 ISSUES ADDRESSED IN DISCRIMINATIVE DICTIONARY LEARNING

The existing approaches to obtain discriminative dictionary are an attempt to improve discriminability of dictionary for classification task. The progress of attaining discriminative dictionary is still in its infancy, a long way to go. In dictionary learning, it inherently contains redundant dictionary atoms which improve sparsity while reconstructing signals. But in the context of discriminative learning, redundant dictionary atoms can be removed while retaining discriminative atoms. We address this problem by incorporating information bottleneck approach to remove redundant dictionary atoms. This approach not only provides discriminative dictionary, but also gives compact dictionary which led to the computational efficiency of classification tasks.

Kernelization is also introduced to improve discriminability among dictionaries. The implementation of kernel dictionary learning is still a challenging task, need to be addressed. To address this issue, we propose an efficient method to incorporate kernelization among dictionaries. In this approach, we introduce a new sampling technique to approximate large kernel matrix. We also exploit underlying coherency among examples to obtain discriminative dictionary. Whenever coherency is high, it ensures compact discriminative dictionary for classification.

2.6 SUMMARY

In this chapter, history of transforms, dictionaries and some of the notable existing approaches in discriminative learning were reviewed. Also, different analytic and adaptive dictionaries are discussed which describe the ability of adaptive learning to represent complex structure of natural signals than analytic approaches. The dictionary learning is the core area in modern signal processing because of its high ability of representation, nature of adapatability and state of the art results. The goal of attaining suitable dictionary for classification is an interesting topic in machine learning community because dictionary learning provides an efficient way for learning. Based on the review over many literatures, the quest for compact and dicriminative dictionary is essential and need to be addressed. In this thesis, we propose novel methods to build compact and discriminative dictionaries for classification.

CHAPTER 3

DICTIONARY LEARNING FOR CLASSIFICATION

Sparse based approaches are widely used in the area of signal processing especially in image and video applications. Object tracking, image charecterization, image denoising, video super resolution, face hallucination, image quality assessment, action recognition are some of the fields where sparse representation has been extensively used. The sparse representation reconstructs the input signal using linear combination of sparse set of fundamental units, aka *atoms*, which are often grouped into a structure called *dictionary* [83] [84]. It is preferred to have overcomplete dictionary which results better representation while having more sparsity. In this overcomplete dictionary, there are more number of unknowns than equations so the signal can have more than one representations.

In this chapter, we discuss naive classification approach using dictionary learning. The Section 3.1 describes the learning of dictionary from the input data and Section 3.2 details the labeling of dictionary atoms from atom distribution over the classes. In the Section 3.3, we discuss action videos classification using dictionary learning with two discriminative measures: projection and reconstruction error. The experiments and performance evaluation are detailed in the Section 3.4 and finally, Section 3.5 summarizes the work.

3.1 LEARNING INPUT DATA

We use K-SVD dictionary learning which adaptively learns input data into dictionary and guarantees to converge at local optimum [61]. As we discussed earlier, K-SVD dictionary learning performs sparse coding and dictionary update alternatively. The sparse coding determines non-zero coefficients of dictionary atoms for the reconstruction. The sparse vector $\mathbf{x}$ contains the coefficients for reconstruction and we would prefer the sparse vector of having maximum number of zero coefficients. The constraints are enforced for further reduction of the number of non zero components in the sparse solution. These constraints are either based on reconstruction error or fixed number of sparsity. Here,we have used sparsity based constraint to learn the dictionary. The l_0 norm and l_1 norm are most widely used sparse constraints in many sparse coding algorithms. The K-SVD dictionary learning uses OMP for sparse coding, which uses l_0 norm. In the learning process, the dictionary $\mathbf{D}$ is fixed to obtain sparse matrix $\mathbf{X}$ which minimizes squared error $\|\mathbf{Y} - \mathbf{DX}\|_F^2$ in sparse coding stage, then $\mathbf{X}$ is used to update the dictionary. The optimization function becomes

$$\underset{\mathbf{D},\mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{DX}\|_F^2 \quad \text{subject to} \quad \forall i \ \|\mathbf{x}_i\|_0 \leq T, \tag{3.1}$$

where $\|.\|_0$ denotes l_0 norm, number of non zeros values in sparse vector $\mathbf{x}_i$ restricted to constraint T, and $\|.\|_F$ denotes Frobenius norm which is matrix norm, defined as the sum of the absolute squares of its elements. To update the dictionary $\mathbf{D}$, every column of $\mathbf{D}$ is to be updated and $\mathbf{X}$ is fixed during updation. Each dictionary atom $\mathbf{d}_k$ is to be updated separately, so the updation procedure has to run K times. Here, we learn separate dictionaries for each of the classes and learned dictionaries will be used for classification.

3.2 ATOM DISTRIBUTION AND SHARING

There are two ways to learn the dictionary: one is to obtain single dictionary for all classes of data and another is to learn separate dictionaries for each of the classes. When we learn single dictionary from all classes of input examples, then there is an issue of labeling learned dictionary atoms. The label of each dictionary atom is necessary for classifying test samples. The distribution of each dictionary atom gives important information regarding the class of the atom. Each row in the sparse matrix $\mathbf{X}$ gives the distribution of corresponding dictionary atom among classes, ie., the distribution of sparse coefficients of dictionary atom $\mathbf{d}_k$ lies on the k^{th} row of the sparse matrix. The label of the dictionary atom $\mathbf{d}_k$ can be determined based on the contribution of $\mathbf{d}_k$ in each class and this can be observed from the distribution of dictionary atom $\mathbf{d}_k$ in the sparse matrix. So, the label is assigned based on the maximum contribution of $\mathbf{d}_k$ among different classes in C, i.e.,

$$\underset{c}{\operatorname{argmax}} \sum_{i=1}^{N_c} |\mathbf{x}_{k,i}| \quad , \qquad \forall c \in C \tag{3.2}$$

where $\mathbf{x}_{k,i}$ denotes k^{th} element of sparse vector $\mathbf{x}_i$ and N_c is number of input vectors in class c. In other way, we can say that the maximum amount of class information contained in the dictionary atom determines the label of the dictionary atom. This is maximum a posterior probability of $\mathsf{p}(\mathsf{c}|\mathsf{d}_k)$. Figure 3.1 shows 20 dictionary atoms which are learned from the USPS digit dataset [85] and Table 3.1 shows the corresponding labels obtained using our approach. This clearly shows our approach determines almost correct labels of the dictionary atoms. These dictionary atoms can be shared among different classes if it contributes equally to more than one class which ultimately helps overall recognition task. The sparse distribution of dictionary atoms can be used to compare the similarity among dictionary atoms.

3.3 ACTION VIDEO CLASSIFICATION

Videos are basically time series data. There are many classical approaches for classification of time series data such as hidden markov model (HMM) [86], dynamic time wrapping (DTW) [87], move split merge (MSM) [88], recently deep learning [89, 90]. In [91], Zhang *et al.* work with human action recognition using sparse coding spatial pyramid matching. The Spatio temporal interest points (STIP) from video sequence are projected onto three orthogonal planes to preserve the layout of STIPs. In this work, we use dictionary learning technique to classify action videos.

Fig. 3.1: The dictionary atoms obtained after learning USPS digit dataset

3.3.1 Features

Low level and mid level features are widely used in action recognition. Semantically rich features became more important now a days for the efficient representation of videos. For better motion representation to detect unusual events, Wang and Liu [92] suggested random local feature (RLF) which describes the spatio-temporal information of depth image. Jargalsaikhan [93] *et al.* construct 3D volume along sparse motion trajectories instead of dense trajectories and extract different features like histogram of oriented gradient (HOG), histogram of optical flow (HOF), motion boundary histogram (MBH), trajectory descriptor (TD) etc. to create bag of features (BoF). Wang *et al.* [94] proposed high level concept action unit to represent human actions in videos . In this, authors proposed context-aware spatial-temporal descriptor to improve the discriminability of the traditionally used local spatial-temporal descriptors and based on the statistics from this descriptor, action unit is derived from the context aware descriptor using graph regularized non-negative matrix factorization, which provides more geometrical information. Action bank, a high level representation of videos, which consists of output of many action detectors that gives a correlation volume. In

Table 3.1: The labels obtained using our approach, which are corresponds to dictionary atoms in Figure 3.1

4	1	7	0	3
0	5	2	7	9
0	1	0	3	0
8	6	8	4	2

this work, we have used action bank features which have been proposed by Sadanand and Corso in their work [95].

3.3.2 Classification approach

The learned dictionaries from K-SVD learning are used for classification of action videos. Two different measures, reconstruction error and projection, are applied to get discriminative information for this classification task. The dictionary $\mathbf{D}$ is obtained by concatenation of all learned dictionaries of each action video category. Let us consider there are m action categories, then the dictionary $\mathbf{D}$ becomes

$$\mathbf{D} = [\mathbf{d}_{1,1} \dots \mathbf{d}_{1,n}, \mathbf{d}_{2,1} \dots \mathbf{d}_{2,n} \dots\dots \mathbf{d}_{m,1} \dots \mathbf{d}_{m,n}].$$

The learned dictionary $\mathbf{D}_k = [\mathbf{d}_{k,1} \dots \mathbf{d}_{k,n}]$ denotes k^{th} action category which contains n column vectors or dictionary atoms. The test vector $\mathbf{y}$ can be approximated as a linear combination of few atoms over the dictionary $\mathbf{D}_k$ of each action category, $\mathbf{y} \approx \mathbf{D}_k\mathbf{x}_k$, the sparse vector $\mathbf{x}_k$ contains coefficients of the dictionary atoms in k^{th} dictionary $\mathbf{D}_k$

for the reconstruction of test vector $\mathbf{y}$. Here, the sparse coding algorithm OMP is used to obtain sparse vector $\mathbf{x}_k$ for the dictionary $\mathbf{D}_k$ using the test vector $\mathbf{y}$

$$\mathbf{x}_k = \mathrm{OMP}(\mathbf{D}_k, \mathbf{y}, T), \qquad k = 1 \ldots m \tag{3.3}$$

where T is the sparse constraint. Now we can find reconstruction error $\mathbf{r}_k$ of k^{th} action category for the test vector $\mathbf{y}$ using dictionary $\mathbf{D}_k$ and corresponding sparse vector $\mathbf{x}_k$. Then the reconstruction error $\mathbf{r}_k$ becomes

$$\mathbf{r}_k = \|\mathbf{y} - \mathbf{D}_k \mathbf{x}_k\|_2^2. \tag{3.4}$$

Then we can form reconstruction error vector $\mathbf{r} = [r_1, r_2, \ldots, r_m]^T$ which contains reconstruction errors of $\mathbf{y}$ from m dictionaries. The minimum reconstruction error determines action category of the test vector $\mathbf{y}$. Projection is another discriminative measure we used here for classification. The test vector $\mathbf{y}$ is projected on to each of the dictionaries for the classification of action videos. The projection matrix $\mathbf{P}_k$ of k^{th} dictionary $\mathbf{D}_k$ is constructed as

$$\mathbf{P}_k = \mathbf{D}_k (\mathbf{D}_k^T \mathbf{D}_k)^{-1} \mathbf{D}_k^T. \tag{3.5}$$

This projection matrix $\mathbf{P}_i$ is used to project test vector $\mathbf{y}$ onto the dictionary $\mathbf{D}_i$. Then norm of the projection of test vector $\mathbf{y}$ can be considered as discriminative measure for the classification. The norm $\mathbf{p}_k$ of projection of $\mathbf{y}$ on the dictionary atoms in $\mathbf{D}_k$ is

$$p_k = \|\mathbf{P}_i \, \mathbf{y}\|_2. \tag{3.6}$$

Similar to reconstruction error, we can form projection vector $\mathbf{p} = [p_1, p_2, \ldots, p_m]^T$ contains norms of projection of $\mathbf{y}$ onto m dictionaries. The maximum projection indicates more correlation of test vector $\mathbf{y}$ to the vector space generated by the dictionary atoms in the corresponding dictionary. This ultimately gives the action category of test vector $\mathbf{y}$.

We can use both reconstruction error and projection together for classification by assigning weights to them, so that, we can utilize the advantages of both discriminative measures to improve the classification. For this purpose, the reconstruction vector **r** to be sorted in ascending order because minimum reconstruction gives the class information. Similarly, projection vector **p** is to be sorted in descending order because maximum projection gives class information. The weights are assigned such that lowest reconstruction error and highest projection are awarded maximum weights. Then, the final score is calculated by adding corresponding weights of each action category for decision making in classification. Suppose we have 5 action categories: action A, action B, action C, action D, and action E, then the corresponding reconstruction error vector $\mathbf{r} = [r_A, r_B, r_C, r_D, r_E]^T$ and projection vector $\mathbf{p} = [p_A, p_B, p_C, p_D, p_E]^T$. After sorting reconstruction vector **r** in ascending order and projection vector **p** in descending order, the weights are assigned to both **r** and **p** as shown in the Table 3.2.

Table 3.2: Weights given to both reconstruction error vector **r** and projection vector **p**

r	weightage	**p**	weightage
r_B	5	p_D	5
r_C	4	p_C	4
r_E	3	p_E	3
r_D	2	p_B	2
r_A	1	p_A	1

Then the final score of each class can be determined by adding corresponding weights as shown in Table 3.3. The action category which is having maximum score will be assigned to test vector **y**. In the above example, test vector **y** belongs to action category C. This approach tries to reduce error occurring in reconstruction error and projection. The intuition is that, the actual action category of test vector will always reside among top of the sorted vectors of **r** and **p**.

Table 3.3: Final weights assigned to each action category

Category	Final score
A	2
B	7
C	8
D	7
E	6

3.4 EXPERIMENTAL RESULTS

In our experiment, action videos are classified in 3 ways, namely, reconstruction error based, projection based, and weights given to both reconstruction error and projection. In reconstruction error based method, action category belonging to minimum reconstruction error is assigned to test video. In projection based method, action category belonging to maximum projection is assigned to test video. In the third method, total score is calculated as explained in section 3.3.2 and then action category belonging to maximum score is assigned to test video. The experiments are conducted with standard action datasets KTH [96], UCF50 [97] and HMDB51 [98]. The UCF50 and HMDB51 are more challenging and realistic dataset compared to KTH action dataset. For each action category, the dictionary has been learned by K-SVD dictionary learning. All results are taken as the average of 5 iterations and size of the learned dictionary and sparsity constraint are determined empirically. We achieved comaprably better results as shown in Table 3.4. Action bank [95] is used as feature vector for the dictionary.

3.4.1 Evaluation on KTH action dataset

In this dataset, there are 25 different subjects performing 6 different actions, which are walking, jogging, running, boxing, hand waving, and hand clapping. The data are partitioned into 3 folds: 2 folds used as training data, remaining one as testing data. The size of the learned dictionary is set to 20% of training data and we considered sparsity

Table 3.4: Overall classification performance (figures in %)

Classifier	KTH	UCF50	HMDB51
SVM [95]	98.20	57.90	26.90
Reconstruction Error	97.22	55.74	22.64
Projection	**97.69**	**59.30**	18.60
Weighted method	97.22	56.49	**23.62**

constraint as $T = 5$. We obtained the performance accuracy of 97.7% (benchmark is 98.2% [95]) which is reasonably good when compared to benchmark result. As shown in Table 3.5, all action videos belong to boxing, jogging, running, and walking are correctly classified. In clapping and handwaving, few videos are misclassified because there is lot of similarity between clapping and handwaving actions.

Table 3.5: KTH dataset: Confusion matrix of performance

	boxing	clapping	handwaving	jogging	running	walking
boxing	1	0	0	0	0	0
clapping	0	0.94	0.06	0	0	0
handwaving	0	0.08	0.92	0	0	0
jogging	0	0	0	1	0	0
running	0	0	0	0	1	0
walking	0	0	0	0	0	1

3.4.2 Evaluation on UCF50 action dataset

This is one of the challenging action datasets. There are 50 action categories and 6950 action videos in all categories. There are 25 persons performing actions in each category. In this experiment, $2/3^{\text{rd}}$ of action videos are considered for training and remaining for testing. There are 50 dictionary atoms learned from each of the action categories and set the sparsity constraint $T = 5$. Here, we could achieve the classification accuracy of 59.3% (benchmark is 57.9% [95]) which is better than the benchmark result. The detailed classification results of each action category is shown in Table 3.6 and some of the actions such as punch, billiards, jumping jack, bench press are showing good results.

Table 3.6: UCF50: Performance in each of the action categories in sorted order

Punch	0.96	HulaHoop	0.68	JugglingBalls	0.47
Billiards	0.94	Drumming	0.68	Swing	0.47
JumpingJack	0.93	Fencing	0.68	BaseballPitch	0.46
BenchPress	0.89	Kayaking	0.67	TennisSwing	0.45
HorseRiding	0.88	PullUps	0.63	VolleyballSpiking	0.45
HorseRace	0.86	Basketball	0.62	PlayingViolin	0.42
ThrowDiscus	0.84	Nunchucks	0.62	PizzaTossing	0.42
Mixing	0.83	HighJump	0.61	Biking	0.42
JumpRope	0.80	PushUps	0.57	SalsaSpin	0.41
RockClimbingIndoor	0.80	PlayingTabla	0.56	Diving	0.41
SkateBoarding	0.78	TaiChi	0.55	RopeClimbing	0.35
PlayingGuitar	0.77	MilitaryParade	0.52	PoleVault	0.28
PommelHorse	0.76	JavelinThrow	0.51	WalkingWithDog	0.27
BreastStroke	0.73	SoccerJuggling	0.50	TrampolineJumping	0.26
CleanAndJerk	0.70	YoYo	0.50	Lunges	0.26
GolfSwing	0.70	Rowing	0.49	Skijet	0.15
PlayingPiano	0.69	Skiing	0.48		

3.4.3 Evaluation on HMDB action data

Here, we have conducted experiment with most challenging dataset. There are 51 actions categories and 6766 action videos in this dataset. In this experiment, the dataset is divided into 10 folds in which 9 folds are used for training and remaining one for testing. From each action category, 50 dictionary atoms are learned and sparsity contraint T is set to 5. In this experiment, the classification performance of 23.6% is achieved (benchmark is 26.9% [95]), which is reasonably good result in this dataset. This dataset being a challenging one, we need to extract more discriminative atoms to improve the classification result. The Table 3.7 gives classification result of each action category.

Table 3.7: HMDB51: Performance in each of the action categories in sorted order

catch	0.71	ride_bike	0.32	kick_ball	0.21	eat	0.08
golf	0.60	push	0.32	hug	0.21	climb_stairs	0.08
laugh	0.60	turn	0.31	run	0.18	dive	0.07
walk	0.56	climb	0.31	cartwheel	0.17	sword_exercise	0.07
smile	0.50	talk	0.30	flic_flac	0.17	wave	0.06
pour	0.46	draw_sword	0.29	sit	0.17	shoot_gun	0.03
ride_horse	0.45	hit	0.29	dribble	0.15	somersault	0.02
pullup	0.41	jump	0.28	sword	0.14	kick	0.00
brush_hair	0.40	kiss	0.26	stand	0.14	punch	0.00
situp	0.40	shake_hands	0.26	smoke	0.11	shoot_ball	0.00
pushup	0.35	fencing	0.24	fall_floor	0.09	swing_baseball	0.00
clap	0.35	drink	0.22	pick	0.09	throw	0.00
shoot_bow	0.32	handstand	0.22	chew	0.08		

3.5 SUMMARY

In this chapter, we proposed dictionary learning based classification for action videos with two discriminative measures, reconstruction error and projection. The combination of both discriminative measures can improve overall classification performance. The projection discriminative measure is always not feasible, because the calculation of projection matrix involves the matrix inversion which causes the computational overhead. The more challenging datasets escalate challenges in action video classification. The dictionary learning provides good representation of data and it can be wisely used for classification purpose. Action bank, high level feature, used here to represent videos. Here, we have experimented three approaches for the classification of action videos, viz. reconstruction error based, projection based, and weighted method. Our experiments show that the learned dictionaries can effectively represent action videos and also computationally effective. We can improve performance by building discriminative dictionaries especially for classification tasks.

CHAPTER 4

INFORMATION BOTTLENECK APPROACH FOR COMPACT DISCRIMINATIVE DICTIONARY

The naive classification approach using standard dictionary learning was discussed in previous chapter in which no discriminative dictionaries are considered. In this chapter, an information theoretic approach is proposed to build compact discriminative dictionary for classification tasks by reducing redundancy among atoms in the dictionary. This approach squeezes relevant information with respect to classes for efficient representation, which is referred to as *information bottleneck*. This is a constraint information optimization problem such that mutual information among optimized dictionary and initial dictionary is to be minimized when the constraint of mutual information among class labels and optimized dictionary should be kept minimum. Here, we optimize the dictionary which is learned using standard dictionary learning algorithm. The distribution of dictionary atoms among classes are compared using the distortion measure *Jensen-Shannon divergence* in which adaptive weights are calculated by observing the contribution of dictionary atom throughout the classes. Then the redundant dictionary atoms are removed based on similarity and the final dictionary becomes discriminative and compact, which retains relevant information while keeping less number of atoms. The reconstruction error is used for classification to demonstrate this approach by comparing performance of dictionary, before and after the optimization.

The field of compact and discriminative representation from dictionary or codebook has been extensively addressed and still much relevant in these days [1–5]. This optimization problem has two phases, one is based on sparsity and another is based on the information bottleneck principle. In [5], discrminative and compact representation from visual data are obtained using twenty one binary descriptors and gradient based approaches are used for discriminative tasks. In [99], Chen *et al.* proposed discriminative visual phrase selection for mobile land recognition, in which loss of disriminative information is reduced and commonalities across various categories are removed. For mobile landmark recognition [100], comapct discriminative vocabulary about context information are extracted. The sparse representation and dictionary learning are very powerful tools which are highly applicable in the filed of machine learning. In [101], each of the classes of images is learned as separate dictionaries in which atoms contain common features among classes are shared. Mairal *et al.* learned sparse based discriminative dictionaries [65, 66] for image classification in which all classes are learned together. The important contributions of our proposed approach are: (1) a new information theoretic approach for sparse based classification, (2) the combination of dictionary learning and information bottleneck to build discriminative and compact dictionaries, (3) the use of adaptive weights in the similarity measure *Jensen-Shannon divergence* for the class distribution of each dictionary atom, which determine similarity among atoms.

In this chapter, we discuss an information theoretic approach to obtain discriminative dictionary. In Section 4.1, design of compact and discriminative dictionary using information bottleneck approach is described. The computation of information loss using Jensen-Shannon divergence is detailed in Section 4.2. The Section 4.3 describes removal of redundant dictionary atoms using the proposed approach. In Section 4.4, we have conducted experiments with different datasets to evaluate the efficacy of the proposed approach. Finally, section 4.5 summarizes the work and presents future directions.

4.1 BUILDING COMPACT AND DISCRIMINATIVE DICTIONARY

In this digital world, data growth is scaling up exponentially, so there is a necessity for the efficient represention of visual data. Our goal is the optimization of dictionary such that it maintains maximum discriminative infromation while keeping few number of dictionary atoms for the purpose of classification. Althogh the dictionary learning has been extensively used in signal reconstruction, this powerful tool can be efficiently used for the classification purpose by designing discriminative dictionary. Moreover, the large sized dictionary like overcomplete dictionary for signal reconstruction is not feasible especially in real time machine learning applications such as classification, which claim more computational and memory resources. In this work, first phase of the optimization is to obtain the dictionary $\mathbf{D}$ by training the input data $\mathbf{Y} = [\mathbf{y}_1 \, \mathbf{y}_2 \ldots \mathbf{y}_N]$, then the obtained dictionary is further optimized using information bottleneck principle in the second phase of optimization. The segregation of discriminative dictionary atoms is realized by removing the redundancy among dictionary atoms in $\mathbf{D}$. The removal of redundant dictionary atoms is a difficult task because there is a chance of loosing discriminative information which may degrade the recognition performance. In order to remove this redundancy among atoms in an efficient manner, we utilize information bottleneck principle [102] in which redundant dictionary can be removed while minimizing loss of discriminative information.

The main objective of this work is the extraction of good representative information for discriminative tasks from the input data. In the first phase of optimization, the input data $\mathbf{Y} \in R^{m \times N}$ is optimized or learned into K dictionary atoms by K-SVD dictionary learning. As we have seen in section 2.3.3, the K-SVD dictionary learning uses OMP for sparse coding and SVD for dictionary update to obtain sparse matrix $\mathbf{X} \in R^{K \times N}$ and dictionary $\mathbf{D} \in R^{m \times K}$, respectively. The optimization function is formed as

$$\underset{\mathbf{D},\mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{DX}\|_F^2 \quad \text{subject to} \quad \forall i \;\; \|\mathbf{x}_i\|_0 \leq T, \tag{4.1}$$

where $\|.\|_F$ and $\|.\|_0$ denote Frobenius norm and l_0 norm, respectively, and T is the sparsity constraint imposed on sparse vector $\mathbf{x}$. In the sparse coding stage, the dictionary $\mathbf{D}$ is fixed to obtain sparse matrix $\mathbf{X}$ while minimizing error function $\|\mathbf{Y} - \mathbf{DX}\|_F^2$, whereas for the dictionary update stage, $\mathbf{X}$ is fixed to obtain $\mathbf{D}$. The learned dictionary $\mathbf{D}$ is not meant for discriminative tasks because it contains redundant dictionary atoms which are not necessary for classification. In the next phase of optimization, our aim is to remove redundant dictionary atoms in an efficient manner. In this, the obtained dictionary $\mathbf{D}$ is considered as input and information theoretic approach is used to remove redundancy in $\mathbf{D}$ which is explained in the following section.

4.1.1 Information bottleneck for optimization

Here, the information bottleneck principle is used to remove redundant dictionaries from the learned dictionary $\mathbf{D}$. Tishby *et al.* [103] [102] conceived the idea of information bottleneck in late 1990's. It was an attempt to address the semantic application of information theoretic approach apart from its application flourished in the field of communication during the middle of 20th century. Here, we use *Jensen-Shannon divergence* [17] with adaptive weights to find similarity among dictionary atoms. This is computationally effective similarity measure which results an efficient implementation when compared to similar kind of existing discriminative dictionary learning approaches [2, 3, 78, 79] where calculation of inverse of the matrix consumes much computational complexity.

In this, the main objective is to remove redundant dictionary atoms from the learned dictionary $\mathbf{D}$ which is obtained from K-SVD dictionary training. More precisely, the signal $\mathsf{d} \in \mathrm{D}$ is to be optimized in such a way that the signal d provides information regarding another signal $\mathsf{c} \in \mathrm{C}$. The C and D denote the set notations for class labels and dictionary, respectively. Here, the goal is the compression of dictionary D into $\tilde{\mathrm{D}}$ when keeping as much as information regarding C. For future use, the random variable notations of D, $\tilde{\mathrm{D}}$, and C are as $\mathbb{D}$, $\tilde{\mathbb{D}}$, and $\mathbb{C}$, respectively. As we mentioned, predicting $\mathbb{C}$ from $\tilde{\mathbb{D}}$ should be as close as possible to predicting $\mathbb{C}$ from $\mathbb{D}$, so we need to optimize the rules $\mathbb{D} \to \tilde{\mathbb{D}}$ and $\tilde{\mathbb{D}} \to \mathbb{C}$.

In this constraint information optimization problem, the mutual information between $\tilde{\mathbb{D}}$ and $\mathbb{D}$ to be minimized while constraint of mutual information among $\mathbb{C}$ and $\tilde{\mathbb{D}}$ should keep as maximum as possible. Consider discrete random variable $\tilde{\mathbb{D}}$ from alphabet $\mathcal{D}$, the entropy $H(\tilde{\mathbb{D}})$ becomes

$$H(\tilde{\mathbb{D}}) = -\sum_{\tilde{\mathsf{d}} \in \mathcal{D}} \mathsf{p}(\tilde{\mathsf{d}}) \log \mathsf{p}(\tilde{\mathsf{d}}), \tag{4.2}$$

and $H(\tilde{\mathbb{D}}|\mathbb{D})$ denotes conditional entropy as

$$H(\tilde{\mathbb{D}}|\mathbb{D}) = -\sum_{\tilde{\mathsf{d}}} \sum_{\mathsf{d}} \mathsf{p}(\tilde{\mathsf{d}}, \mathsf{d}) \log \mathsf{p}(\tilde{\mathsf{d}}|\mathsf{d}). \tag{4.3}$$

Then the mutual information $I(\tilde{\mathbb{D}}; \mathbb{D})$ among $\tilde{\mathbb{D}}$ and $\mathbb{D}$ becomes

$$\begin{aligned} I(\tilde{\mathbb{D}}; \mathbb{D}) &= H(\tilde{\mathbb{D}}) - \mathbb{H}(\tilde{\mathbb{D}}|\mathbb{D}) \\ &= \sum_{\tilde{\mathsf{d}}} \sum_{\mathsf{d}} \mathsf{p}(\tilde{\mathsf{d}}, \mathsf{d}) \log \frac{\mathsf{p}(\tilde{\mathsf{d}}, \mathsf{d})}{\mathsf{p}(\tilde{\mathsf{d}})\mathsf{p}(\mathsf{d})} \\ &= \sum_{\tilde{\mathsf{d}}} \sum_{\mathsf{d}} \mathsf{p}(\mathsf{d})\mathsf{p}(\tilde{\mathsf{d}}|\mathsf{d}) \log \frac{\mathsf{p}(\tilde{\mathsf{d}}|\mathsf{d})}{\mathsf{p}(\tilde{\mathsf{d}})}. \end{aligned} \tag{4.4}$$

Information bottleneck approach can be effectively utilized to remove the redundant dictionary atoms. The dictionary atoms in $\tilde{\mathbb{D}}$ form *bottleneck* where the information, that $\mathbb{D}$ contains about $\mathbb{C}$, is squeezed. This can be compared with trade-off between rate and distortion as in the *rate distortion function*, R(D), [104] in which the rate R is focussed on better representation while the distortion D refers compression. Then the tolerable distortion in achievable rate is the important issue to be addressed. In [102, 103], this problem is formulated as a constrained information optimization problem to keep *relevant information* for semantic applications.

To determine discriminative dictionary atoms, the compressed dictionary $\tilde{\mathbb{D}}$ is to be obtained from $\mathbb{D}$ whereas $\tilde{\mathbb{D}}$ should keep maximum information regarding $\mathbb{C}$. The data processing inequality [104] gives Markov chain $\tilde{\mathbb{D}} \to \mathbb{D} \to \mathbb{C}$, which derives the amount of mutual information among $\tilde{\mathbb{D}}$ and $\mathbb{C}$ cannot be greater than original mutual information among $\mathbb{D}$ about $\mathbb{C}$ as

$$I(\tilde{\mathbb{D}};\mathbb{C}) \leq \mathbb{I}(\mathbb{D};\mathbb{C}). \tag{4.5}$$

This optimization problem is formulated such that the constraint $I(\tilde{\mathbb{D}};\mathbb{C})$ should be as high as possible while minimizing the mutual information $I(\tilde{\mathbb{D}};\mathbb{D})$. To solve this problem, now we can formulate the optimization function as

$$\underset{\mathsf{p}(\tilde{\mathsf{d}}),\mathsf{p}(\tilde{\mathsf{d}}|\mathsf{d})}{\operatorname{argmin}} \quad I(\tilde{\mathbb{D}};\mathbb{D}) - \beta\mathbb{I}(\tilde{\mathbb{D}};\mathbb{C}), \tag{4.6}$$

where β indicates the Lagrange multiplier. By minimizing optimization fuction in (4.6), the self consistent equations $\mathsf{p}(\tilde{\mathsf{d}}|\mathsf{d})$ and $\mathsf{p}(\tilde{\mathsf{d}})$ can be obtained . We can solve this problem using a well known iterative procedure called *Blahut-Arimoto Algorithm* [105]. The self consistent equations $\mathsf{p}(\tilde{\mathsf{d}}|\mathsf{d})$ and $\mathsf{p}(\tilde{\mathsf{d}})$ by minimizing mutual information with respect to distortion dist$(\tilde{\mathsf{d}},\mathsf{d})$. The (t+1)th update of this iterative procedure is given by

$$\begin{cases} \mathsf{p}_{\mathsf{t}+1}(\tilde{\mathsf{d}}) & = \sum_{\mathsf{d}} \mathsf{p}(\mathsf{d})\mathsf{p}_{\mathsf{t}}(\tilde{\mathsf{d}}|\mathsf{d}) \\ \mathsf{p}_{\mathsf{t}+1}(\tilde{\mathsf{d}}|\mathsf{d}) & = \frac{\mathsf{p}_{\mathsf{t}}(\tilde{\mathsf{d}})\mathsf{exp}(-\beta\,\mathrm{dist}(\tilde{\mathsf{d}},\mathsf{d}))}{\sum_{\tilde{\mathsf{d}}}\mathsf{p}_{\mathsf{t}}(\tilde{\mathsf{d}})\mathsf{exp}(-\beta\,\mathrm{dist}(\tilde{\mathsf{d}},\mathsf{d}))}. \end{cases} \tag{4.7}$$

These iterations converge to a unique minimum in the convex set of two distributions [104] [105].

The optimal assignments, which minimize (4.6), satisfy the equation,

$$\mathsf{p}(\tilde{\mathsf{d}}|\mathsf{d}) = \frac{\mathsf{p}_{\mathsf{t}}(\tilde{\mathsf{d}})}{\mathcal{N}(\mathsf{d},\beta)}\mathsf{exp}\Bigg[-\beta\sum_{\mathsf{c}}\mathsf{p}(\mathsf{c}|\tilde{\mathsf{d}})\mathsf{log}\frac{\mathsf{p}(\mathsf{c}|\tilde{\mathsf{d}})}{\mathsf{p}(\mathsf{c}|\mathsf{d})}\Bigg], \tag{4.8}$$

where $\mathcal{N}(\mathsf{d},\beta)$ denotes normalization function. The details of this proof can be seen in [103]. The distribution $\mathsf{p}(\mathsf{c}|\tilde{\mathsf{d}})$ is obtained using Markov chain $\tilde{\mathbb{D}} \to \mathbb{D} \to \mathbb{C}$ and Bayes' rule,

$$\begin{aligned} \mathsf{p}(\mathsf{c}|\tilde{\mathsf{d}}) &= \sum_{\mathsf{d}}\mathsf{p}(\mathsf{c}|\mathsf{d})\mathsf{p}(\mathsf{d}|\tilde{\mathsf{d}}) \\ &= \frac{1}{\mathsf{p}(\tilde{\mathsf{d}})}\sum_{\mathsf{d}}\mathsf{p}(\mathsf{c}|\mathsf{d})\mathsf{p}(\tilde{\mathsf{d}}|\mathsf{d})\mathsf{p}(\mathsf{d}) \end{aligned} \tag{4.9}$$

and,

$$\mathsf{p}(\tilde{\mathsf{d}}) = \sum_{\mathsf{d}} \mathsf{p}(\tilde{\mathsf{d}}|\mathsf{d})\mathsf{p}(\mathsf{d}). \tag{4.10}$$

The Kullback-Leibler divergence or relative entropy [16] is a well known similarity measure between two probability distributions. Consider two probability mass functions $\mathsf{q}(\mathsf{x})$ and $\mathsf{p}(\mathsf{x})$, then Kullback-Leibler divergence becomes

$$\mathsf{D}(\mathsf{q}||\mathsf{p}) = \sum_{\mathsf{x}} \mathsf{q}(\mathsf{x})\mathsf{log}\frac{\mathsf{q}(\mathsf{x})}{\mathsf{p}(\mathsf{x})}. \tag{4.11}$$

Now the equation (4.8) can be written as

$$\mathsf{p}(\tilde{\mathsf{d}}|\mathsf{d}) = \frac{\mathsf{p_t}(\tilde{\mathsf{d}})}{\mathcal{N}(\mathsf{d},\beta)}\mathsf{exp}\Big[-\beta\,\mathsf{D}\big(\mathsf{p}(\mathsf{c}|\tilde{\mathsf{d}})||\mathsf{p}(\mathsf{c}|\mathsf{d})\big)\Big]. \tag{4.12}$$

We can notice that the distortion measure in (4.7) is replaced by Kullback-Leibler divergence. It makes sense because this is a natural distortion measure to find distance between distributions $\mathsf{p}(\mathsf{c}|\tilde{\mathsf{d}})$ and $\mathsf{p}(\mathsf{c}|\mathsf{d})$. In this work, we replace Kullback-Leibler divergence with Jensen-Shannon divergence because we can weigh the distributions of class given dictionary atom in the Jensen-Shannon divergence for better comparison and the change in mutual information, δI_c, is also determined in an efficient manner which are explained in the following section.

4.2 JENSEN-SHANNON (JS) DIVERGENCE WITH ADAPTIVE WEIGHTS

As we have seen in section 1.3.4, Jensen-Shannon divergence can be used for more than two distributions and weights can be assigned to each of the distributions. Unlike Kullback-Leibler divergence, JS divergence is symmetric. These properties of Jensen-Shannon divergence are very helpful in our context. In this work, we efficiently merge similar dictionary atoms using Jenson-Shannon divergence and these merging steps are explained in section 4.3. Here, we discuss how to find similar dictionary atoms for merging. From the information bottleneck principle, we obtained the probability distributions in section 4.1.1 which can be used to calculate the mutual information. So, the loss of information can be calculated among dictionary atoms and based on minimum information loss, atoms can be merged. The information loss or change in information, ie. δI_c, can be defined as

$$\delta I_c = I(\mathbb{Z}_m; \mathbb{C}) - I(\mathbb{Z}_{m-1}; \mathbb{C}). \tag{4.13}$$

The information loss is calculated for every possible pair in $\mathbb{Z}_m$ ($\mathbb{Z}_m$ is the current m-partition, each partition consists of dictionary atoms and $\mathbb{Z}_{m-1}$ be the partition after merging a pair in $\mathbb{Z}_m$). It is a greedy way of problem solving where we look for best possible merge for every pair and can find most similar partitions. Using the formula in (4.13), $O(m.|\mathbb{C}|)$ operations are required for each pair. This computation can be improved using the distortion measure Jensen-Shannon divergence, where only $O(|\mathbb{C}|)$ operations are required to calculate mutual information loss after merging process. The mutual information loss, δI_c, can be written in terms of JS divergence [102] as

$$\delta I_c = \big(\mathsf{p}(\mathsf{z_i}) + \mathsf{p}(\mathsf{z_j})\big)\,\mathsf{JS}_\pi\big(\mathsf{p}(\mathsf{c}|\mathsf{z_i}), \mathsf{p}(\mathsf{c}|\mathsf{z_j})\big), \tag{4.14}$$

where weights $\pi = [\pi_i, \pi_j]$. The distributions $\mathsf{p}(\mathsf{c}|\mathsf{z_i})$ and $\mathsf{p}(\mathsf{c}|\mathsf{z_j})$ are assigned weights π_i and π_j, respectively. In this, weights are adaptive, which depends probability of corresponding dictionary atom or partition. The values of these adaptive weights are obtained based on the contribution of dictionary atom among different classes. Here, we give more weightage to distribution such that the dictionary atom is used by maximum number input samples in all classes. The weights π_i and π_j can be formulated as

$$\begin{aligned} \pi_i &= \frac{\mathsf{p}(\mathsf{z_i})}{\mathsf{p}(\mathsf{z_i}) + \mathsf{p}(\mathsf{z_j})} \\ \pi_j &= \frac{\mathsf{p}(\mathsf{z_j})}{\mathsf{p}(\mathsf{z_i}) + \mathsf{p}(\mathsf{z_j})} \end{aligned} \tag{4.15}$$

The JS divergence is computationally effective to determine information loss by comparing the distributions $\mathsf{p}(\mathsf{c}|\mathsf{z_i})$ and $\mathsf{p}(\mathsf{c}|\mathsf{z_j})$, which will detect most similar dictionary atoms. The loss of mutual information, δI_c, is depicted in figure 4.1 during the removal of dictionary atoms. As you can observe in the figure, the information loss raised quickly after particular point where we can stop the process of removing dictionary atoms. It can be seen that the loss of information increases rapidly after a particular point where we can stop the removal of redundant dictionary atoms. From the figure 4.1, it can be concluded that we can decide the optimal number of dictionary atoms in the final dictionary by observing loss of information.

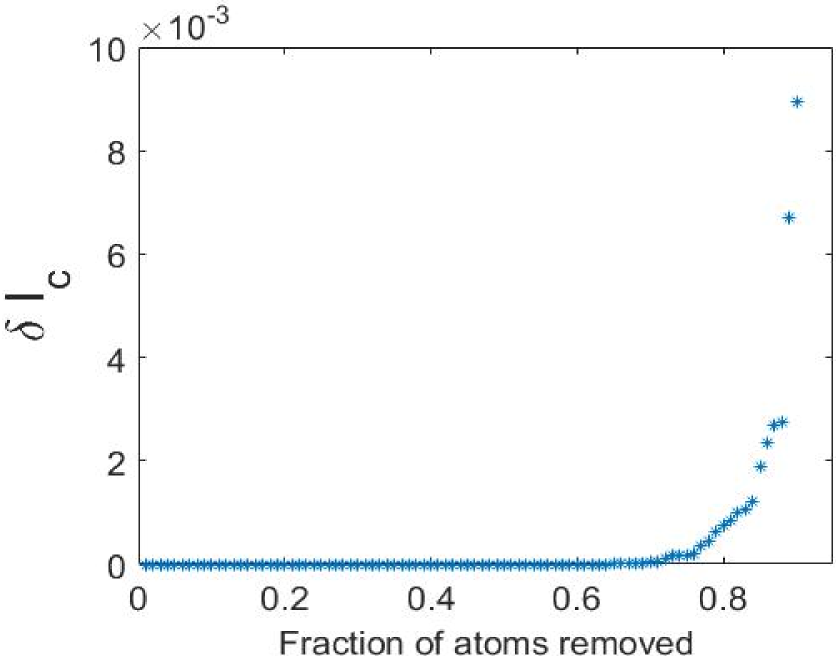


Fig. 4.1: KTH dataset: Information loss, δI_c, during the removal of dictionary atoms.

4.3 REMOVAL OF REDUNDANT DICTIONARY ATOMS

In order to remove redundancy among dictionary atoms, we need to merge similar dictionary atoms obtained in section 4.2 using *Jensen-Shannon divergence.* In this section, we discuss merging process for the removal of atoms and updating of probability distribution after the removal redundant atoms. The self consistent equations are taken based on agglomerative information bottleneck principle [102] in which β in (4.8) become ∞. Here, we use one new variable $\mathbb{Z}$ to avoid confusion and variable $\mathbb{Z}$ is initialized with $\mathbb{D}$. In this, the relation between $\mathbb{Z}$ and $\tilde{\mathbb{Z}}$ is just one step away in the process of merging, i.e., compressed representation $\tilde{\mathbb{Z}}$ is obtained after merging atoms in $\mathbb{Z}$. To merge dictionary atoms, the initialization becomes

$$\mathbb{Z} = \mathbb{D}, \; \mathsf{z_i} = \mathsf{d_i} \tag{4.16}$$

$$\mathsf{p(c|z_i)} = \mathsf{p(c|d_i)} \text{ for every } \mathsf{c} \in \mathbb{C}, \tag{4.17}$$

$$\mathsf{p(z_i|d_j)} = \begin{cases} 1 & \text{if j=i} \\ 0 & \text{otherwise} \end{cases} \tag{4.18}$$

and compute distances for every $i, j \in \{1, \ldots, N\}, i < j$

$$\mathbf{S}_{i,j} = \big(\mathsf{p(z_i)} + \mathsf{p(z_j)}\big)\, \mathsf{JS}_\pi\big[\mathsf{p(c|z_i)}, \mathsf{p(c|z_j)}\big] \tag{4.19}$$

The distance matrix $\mathbf{S}$ is a lower triangular matrix and lowest entry in the matrix determine similar atoms in the process of merging. The updation of corresponding probabilities are to be carried out after merging dictionary atoms. Here, at a time, two similar atoms are merged rather than more than two dictionary atoms. This helps to understand the information loss at each step of merging and we can take decision regarding the optimal number dictionary atoms in the final dictionary. In the process of merging, the redundant atoms are removed with minimum discriminative information loss. This can be easily obtained from the distance matrix $\mathbf{S}$ as

$$< u, v >= \operatorname*{argmin}_{i,j}(\mathbf{S}_{i,j}), \tag{4.20}$$

and merge atoms as $(z_u, z_v) \to \tilde{z}$. After merge, the probabilities related to merged dictionary atom, $\tilde{z}$, is to be updated as

$$\mathsf{p}(\tilde{\mathsf{z}}) = \mathsf{p}(\mathsf{z_u}) + \mathsf{p}(\mathsf{z_v}) \tag{4.21}$$

$$\mathsf{p}(\mathsf{c}|\tilde{\mathsf{z}}) = \frac{1}{\mathsf{p}(\tilde{\mathsf{z}})}\big(\mathsf{p}(\mathsf{z_u}, \mathsf{c}) + \mathsf{p}(\mathsf{z_v}, \mathsf{c})\big) \tag{4.22}$$

$$\mathsf{p}(\tilde{\mathsf{z}}|\mathsf{d_j}) = \begin{cases} 1 & \text{if } \mathsf{d_j} \in \tilde{\mathsf{z}} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{Z} = \Big\{\mathrm{Z} - \{z_u, z_v\}\Big\} \cup \Big\{\tilde{z}\Big\} \tag{4.23}$$

The distance matrix $\mathbf{S}$ is to be updated such that distance between $\tilde{z}$ and remaining z_i's are to be inserted and entries correspond to z_u and z_v are to be removed. This approach gives an efficient way of removing redundancy in the learned dictionary and the process of merging can be stopped at the point where the information loss, δI_c, is as minimum as possible. This helps to approximate the minimum number of dictionary atoms to be retained without loosing much discriminative information. One representative is atom to be selected as the mean of grouped dictionary atoms from each merged group. Next we conduct experiments to validate how good this optimized dictionary is.

4.4 EXPERIMENTAL RESULTS

We use different benchmark datasets for the evaluation of the proposed optimization approach. For the experiment, we have used USPS digit database [85], AR face database [106] and three action datasets, namely, UCF sports [97], KTH [96], and HMDB51 [98]. The action datasets are represented by action bank features which are used by Sadanand and Corso in their work [95]. The action bank features comprise of many individual action detectors which constitute mid-level representation of action data and carry rich semantic information. For all databases, feature vectors are stacked as matrix. Moreover, each feature vector is mean extracted and normalized to unit l_2 norm. The K-SVD dictionary learning is used to obtain initial dictionary. In this experiment, we have performed 20 dictionary learning iterations and the sparsity constraint T is determined empirically.

Information bottleneck approach is used for the further optimization of learned dictionary as described in section 4.1.1 and this optimized dictionary is used in the experimental evaluation. In [3], the learned dictionary is optimized by comparing sparse decompositions in terms of mutual information using Guassian process. In this, inverse of covariance of sparse matrix is to be determined which is computationally expensive. In our method, instead of computing inverse of the matrix, we used computationally efficient Jensen-Shannon divergence to compare distributions as explained earlier. The recognition accuracies are determined based on the minimum reconstruction error as discussed in the chapter 3. We also compare our approach with traditional classifiers such as K nearest neighbor (KNN), support vector machine (SVM) etc. All experiments are conducted on the same machine and execution time of classification and dictionary optimization are determined to compare with other similar approaches.

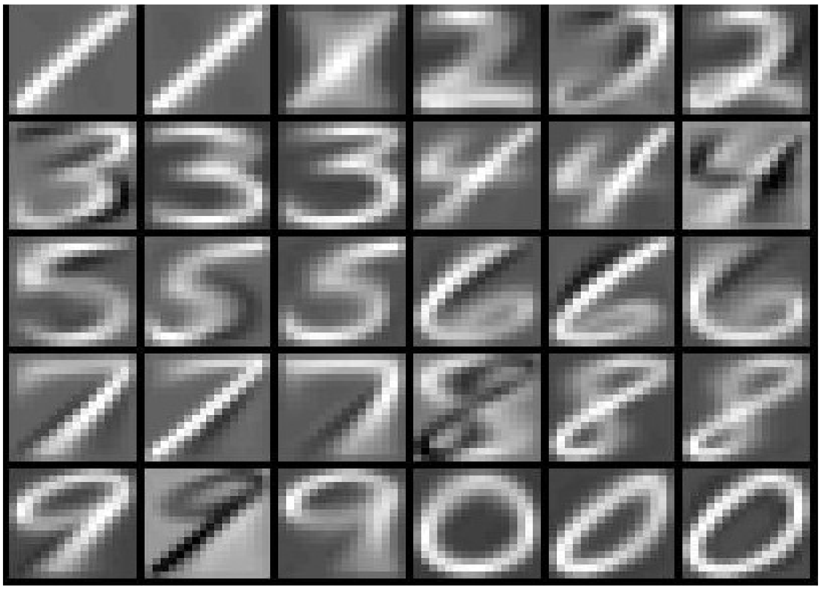


Fig. 4.2: Standard K-SVD algorithms is applied to obtain dictionary atoms of USPS digit dataset

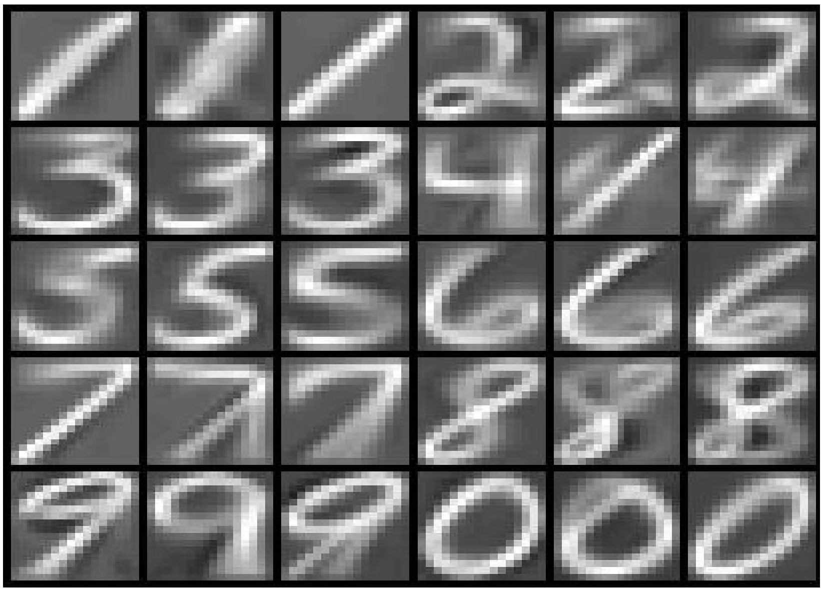


Fig. 4.3: Proposed approach is applied to obtain dictionary atoms of USPS digit dataset

4.4.1 Evaluation on USPS digit dataset

The USPS database consists of handwritten digits of 0-9 which constitute 10 classes. There are 7291 training and 2007 test images of digits of size 16 $\times$ 16 which become feature vector of dimension 256. The figures 4.2 and 4.3 compare dictionary atoms obtained directly and proposed approach. The figure 4.2 gives visualization of dictionary atoms obtained using the direct application of K-SVD dictionary learning on USPS data. Whereas figure 4.3 visualizes dictionary atoms obtained after removing dictionary atoms using proposed approach from the initial dictionary of size 100. It can be observed that atoms in figure 4.3 are more discriminative than figure 4.2 which shows our optimization method tries to retain maximum discriminative atoms than direct approach.

First, we evaluate the removal of dictionary atoms does not affect classification accuracy. For the experiment, 40, 30 and 10 dictionary atoms are learned from each class which constitute dictionary of size 400, 300, and 100, respectively. The sparsity constraint T is taken as 5. Table 4.1 shows classification accuracy and time of the initial dictionary and optimized ditionary in which it preserves the accuracy even after removing redundant dictionary atoms. The maximum performance we achieved is 97.2% which is comparable to other approaches [12]. Table 4.2 compares the classification accuracy and time with other traditional approaches. Our approach shows good computational efficiency in classification when compared to SVM and KNN. Another impact of our approach is the time taken for the optimization process. We compare our method with other similar methods MMI, MMI-1 [3], Table 4.3 shows proposed approach clearly outperforms other methods in computational aspects. Table 4.4 indicates adaptive weightages help to merge similar dictionary atoms compared to equal weghtages (at a time only two distributions are compared, so weights are 0.5 and 0.5) and this adaptive weights improve overall accuracy.

4.4.2 Evaluation on AR face dataset

The original AR Face database contains 4000 color images of faces from 126 people, namely, 70 men and 56 women. The frontal view face images are taken based on different facial expressions, illumination conditions, occlusions etc. Following the experiments in [76], 2600 images were chosen from first 50 classes of males and first 50 classes of females, so total 100 classes for the experiment. Each class has 26 images in which 20 for training and remaining for testing. As you can see in Table 4.1, dictionary is learned with the size of 1500 atoms because the number of classes are high and we obtained 94.6% accuracy which is comparable to [12, 76]. The atom removal from dictionary of size 800 causes much performance degradation due to loss of more discriminative information. Table 4.2 gives performance comparison of the proposed method with KNN and SVM. It can be observed that the proposed dictionary learning method performs better than KNN and SVM in terms of both classification accuracy and time. As shown in Table 4.4, the adaptive weightages improve the classification performance significantly.

Table 4.1: The comparison between initial dictionary and optimized dictionary in terms of recognition accuracy (%) and time (sec.)

	Initial	accuracy	time	Optimized	accuracy	time
USPS	$\|D\| = 400$	97.20	0.124	$\|D\| = 300$	96.80	0.118
	$\|D\| = 300$	95.50	0.119	$\|D\| = 200$	95.20	0.094
	$\|D\| = 100$	92.20	0.086	$\|D\| = 90$	92.60	0.069
AR	$\|D\| = 1500$	94.60	1.526	$\|D\| = 1400$	93.00	1.420
	$\|D\| = 1000$	92.10	1.350	$\|D\| = 900$	90.50	1.263
	$\|D\| = 800$	89.00	1.116	$\|D\| = 700$	82.83	1.031
UCF10	$\|D\| = 100$	95.60	0.194	$\|D\| = 70$	95.00	0.130
	$\|D\| = 80$	87.20	0.166	$\|D\| = 70$	88.00	0.120
	$\|D\| = 60$	84.00	0.154	$\|D\| = 50$	84.20	0.117
KTH	$\|D\| = 300$	96.30	0.708	$\|D\| = 200$	97.60	0.542
	$\|D\| = 200$	94.51	0.555	$\|D\| = 100$	94.53	0.344
	$\|D\| = 100$	94.41	0.343	$\|D\| = 50$	94.26	0.269
HMDB 51	$\|D\| = 900$	36.70	195.068	$\|D\| = 600$	32.30	87.550
	$\|D\| = 650$	33.32	90.253	$\|D\| = 590$	32.57	85.931

4.4.3 Evaluation on UCF sports action data

The UCF sports action dataset has 10 different classes of sports viz. diving, golfing, kicking, weight lifting, horse riding, running. skate boarding, swinging bench, swinging side angle and walking. Experiments have been done with five fold cross validation, ie., four folds are used for training and remaining one for testing. We experiment different initial dictionaries of size 100,80, 60 with sparsity of 3, 10, 15, respectively. The dictionary of size 60 learned with sparsity $T = 15$, this includes more dictionary atoms while learning and improves overall recognition performance. The atoms are removed in each iteration and our results are compared with random removal, MMI, MMI-2 shown in Figure 6.2. Whenever it reaches smaller and smaller dictionary size, our method clearly

Table 4.2: Comparing proposed method with KNN and linear-SVM classifier in terms of recognition accuracy (%) and testing time (sec.)

	KNN		SVM		**Proposed Method**	
	Acc.	Time	Acc.	Time	Acc.	Time
USPS	91.00	1.212	95.00	1.961	**97.60**	**0.512**
AR Face	85.00	0.204	91.00	0.295	**94.60**	**0.193**
UCF10	88.00	0.312	95.00	0.486	**95.60**	**0.203**
KTH	78.95	1.950	97.15	3.121	**97.60**	**1.942**
HMDB 51	26.59	190.12	26.91	450.61	**35.32**	**188.190**

outperforms other methods. After removing 50% of atoms from the initial dictionary, the proposed method still maintain good performance. The recognition accuracies of initial dictionary and optimized dictionary are shown in Table 4.1 which indicate our method could remove the redundant dictionary atoms without degrading recognition performance. This resulted in better classification time. The dictionaries of size 80 and 60 slightly improve the recognition accuracy after removing the redundancy. In addition, this optimization tremendously reduces classification time compared to other traditional approaches such as SVM, KNN as shown in Table 4.2. Our approach shows better performance in both recognition accuracy and testing time compared to SVM and KNN classifier. The computational efficiency of our approach is also better than MMI and MMI-2 as shown Table 4.3. The performance of our proposed approach with other state of art approach is shown in Table 4.5 and we obtained comparable result with [95], but dominate performances in other methods [3] [107] [108] [109].

The figure 4.5(a) shows mutual information between optimized dictionary $\tilde{\mathbb{Z}}$ and class $\mathbb{C}$, ie.,$I(\tilde{\mathbb{Z}};\mathbb{C})$. It can be observed that, our optimization problem tries to maximize $I(\tilde{\mathbb{Z}};\mathbb{C})$. In contrast to $I(\tilde{\mathbb{Z}};\mathbb{C})$, the mutual information between optimized dictionary $\tilde{\mathbb{Z}}$ and initial dictionary $\mathbb{D}$, $I(\tilde{\mathbb{Z}};\mathbb{D})$, is to be minimized which can be seen in figure 4.5(b).

Table 4.3: Comparing the computational efficiency (measured in seconds) of the proposed approach with other methods, namely, MMI, MMI-2.

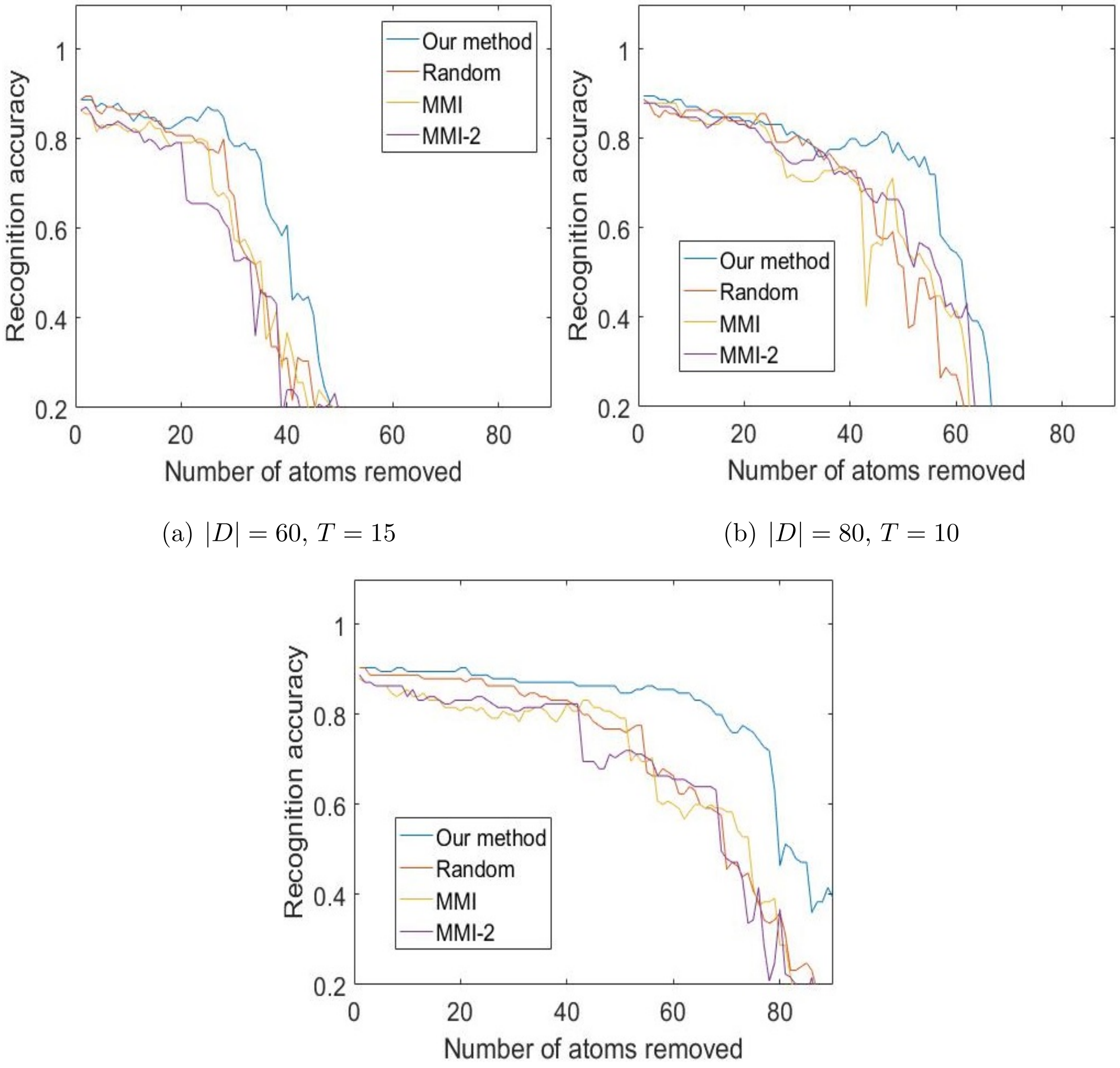
	Initial dictionary size	Optimized dictionary size	MMI	MMI-2	Our method
UCF	100	50	0.74	0.70	0.67
KTH	200	100	5.85	6.64	1.90
KTH	300	150	14.43	15.95	4.32
USPS	400	300	15.21	16.27	4.15

Table 4.4: The recognition performance comparison when we use equal weights and adaptive weights.

	Equal wts.	Adaptive wts.
USPS	96.30	97.20
AR Face	92.10	94.55
UCF10	94.10	95.60

4.4.4 Evaluation on KTH action dataset

In this dataset, 25 different subjects performing 6 different actions, which are walking, jogging, running, boxing, hand waving and hand clapping. We partitioned data into 3 folds and 2 folds used as training data, remaining one as testing data. Here, three different initial dictionaries of sizes 300, 200, 100 are learned with sparsity 3, 7, 3, respectively. The Table 4.1 compares recognition accuracies of initial and optimized dictionaries on different dictionary sizes. Consider the dictionary of size 200, after removing half of the dictionary still it shows good accuracy. The Table 4.2 compares



(a) $|D| = 60$, $T = 15$

(b) $|D| = 80$, $T = 10$

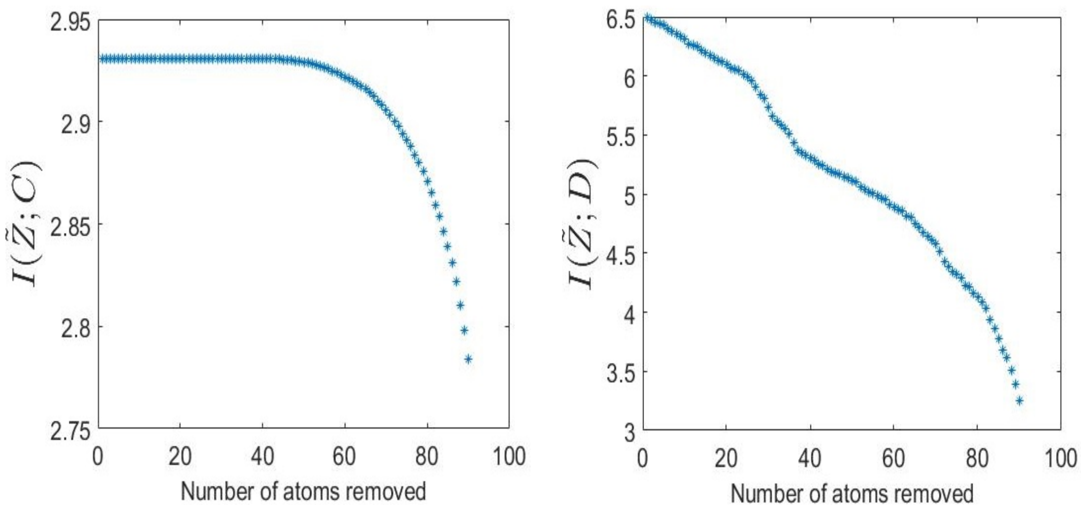
(c) $|D| = 100$, $T = 2$

Fig. 4.4: Comparing recognition performances of the proposed approach (for different dictionary sizes) with other methods, namely, random removal of atoms, MMI and MMI-2 using UCF action dataset.

recogniton accuracy and testing time with other approaches like KNN and SVM. We have achieved good recognition accuracy and comparable testing time when compared to KNN. In case of SVM, we have better testing time and comparable recognition performance. This shows our proposed approach can achieve good recognition accuracy while maintaining good testing time. As shown in Table 4.3, computational time of

Table 4.5: Comparing Performance of UCF sports action classification with state of the arts.

Method	Average performance (%)
Proposed method	**95.6**
Sadanand et.al [95]	95.0
Yao et al. [109]	86.6
Qiu et al. [3]	83.6
Rodriguez et al. [108]	69.2
Yeffet Wolf [107]	79.2



(a) (b)

Fig. 4.5: Mutual information $I(\tilde{\mathbb{Z}};\mathbb{C})$ and $I(\tilde{\mathbb{Z}};\mathbb{D})$ when removing dictionary atoms in UCF.

our optimization is better than other approaches which suffer computational burden of inverse calculation of the matrix. We achieved recognition accuracy of 97.60% which is comparable to 98.20% in [95]. Figure 4.6 shows comparison of our result with random removal, MMI and MMI-2. In this dataset, performance of all methods differs slightly, because this is comparatively easy dataset and feature vectors are well represented.

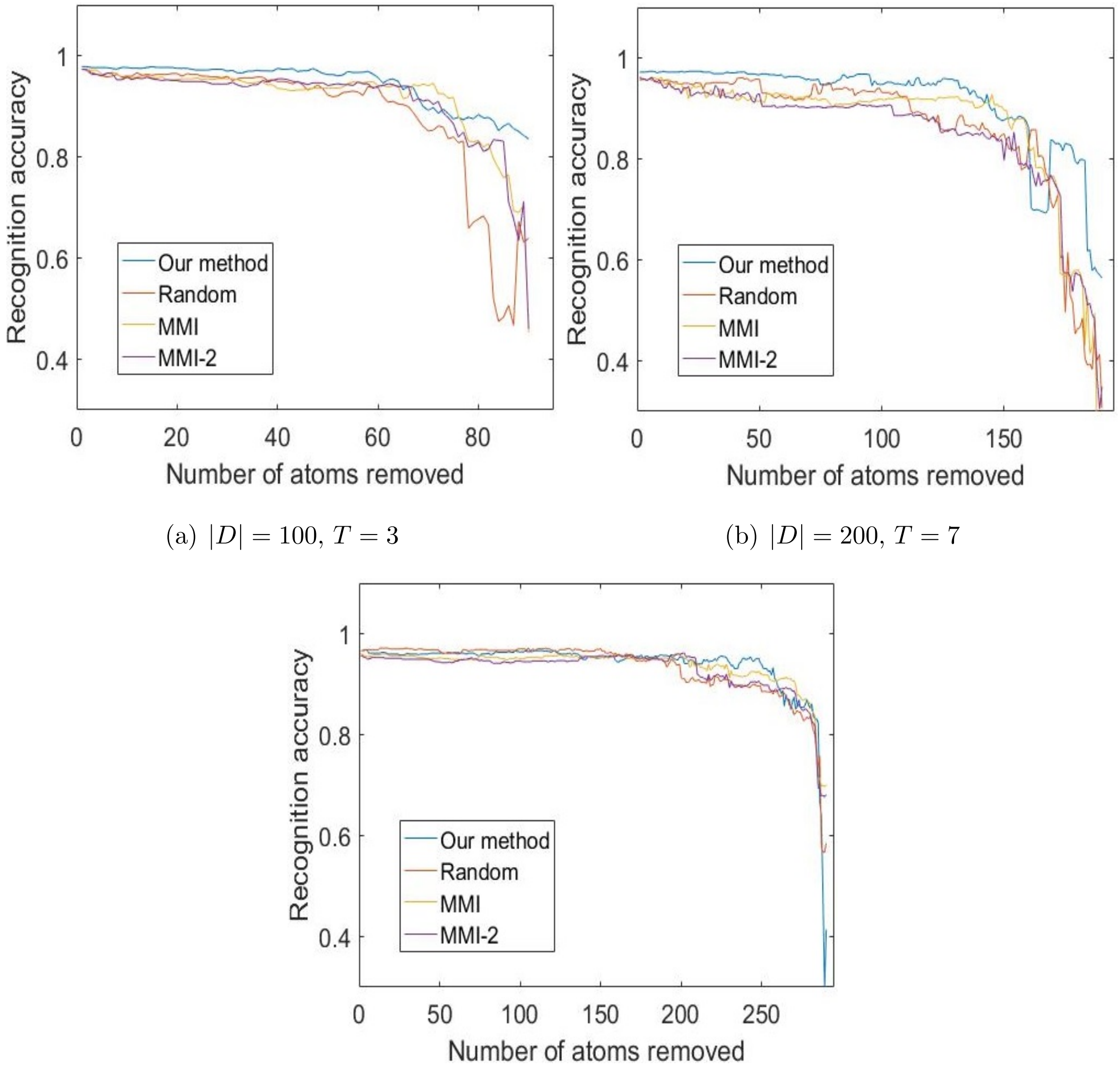
Still the clear difference is evident at smaller dictionary sizes as seen in Figure 4.6. Two confusion matrices of dictionary of size 100 and its optimized dictionary of size 50 using our method are shown in the Table 4.6 and 4.7, respectively. It can be observed that there is a minute variation in the recognition performance which clearly indicates that this proposed method retains maximum discriminative information while optimizing.

Table 4.6: KTH dataset: Confusion matrix using initial dictionary size of 100

	boxing	clapping	handwaving	jogging	running	walking
boxing	1.0	0	0	0	0	0
clapping	0	0.92	0.08	0	0	0
handwaving	0	0.03	0.97	0	0	0
jogging	0	0	0	1.0	0	0
running	0	0	0	0	1.0	0
walking	0	0	0	0	0	1.0

Table 4.7: KTH dataset: Confusion matrix using optimized dictionary size of 50

	boxing	clapping	handwaving	jogging	running	walking
boxing	1.0	0	0	0	0	0
clapping	0	0.92	0.06	0.02	0	0
handwaving	0	0.06	0.94	0	0	0
jogging	0	0	0	1.0	0	0
running	0	0	0	0	1.0	0
walking	0	0	0	0	0	1.0



(a) $|D| = 100$, $T = 3$

(b) $|D| = 200$, $T = 7$

(c) $|D| = 300$, $T = 1$

Fig. 4.6: Comparing recognition performances of the proposed approach (for different dictionary sizes) with other methods, namely, random removal of atoms, MMI and MMI-2 using KTH action dataset.

4.4.5 Evaluation on HMDB action data

Here, we conducted experiment with very challenging dataset discussed in previous sections. There are 51 actions categories in this dataset. In this experiment, the dataset is divided into 10 folds in which 9 folds are used for training and remaining one for testing. We achieved recognition accuracy of 36.70% compared to 26.9% [95]

which is benchmark result using action bank features. The recognition accuracy and computational time of initial and optimized dictionaries are shown in Table 4.1. We have learned dictionaries of size 900 and 650 with sparsity T=10. The dictionary of size 650 is optimized into 590 dictionary by removing 60 atoms, but recognition accuracy only vary from 35.32% to 35.17%. There are 300 atoms removed from the dictionary of size 900 and it can be seen that the recognition accuracy is reduced to 4.4% in the optimized dictionary, but computational time reduced drastically. There is more information loss in this compared to previous dataset because of the high variability and large number of classes in the dataset, but still it gives comparable performance. The Table 4.2 compares the proposed method with KNN and SVM in which the time taken for SVM classifier is more than double of testing time of our method because of the large input data. We have achieved the recognition performance of 35.32% compared to 26.59% of KNN.

4.5 SUMMARY

In this chapter, we proposed an efficient approach to build compact and discriminative dictionary using an information theoretic approach. Dictionary learning is the fastest way to get initial dictionary rather than clustering approach used in previous approaches [4] [80]. In this work, we formulated constraint information optimization problem, which is motivated from information bottleneck approach, to obtain compact discriminative dictionary. Using this approach, we remove redundant atoms with the help of *Jensen-Shannon divergence* which is simple and computationally effective way to find similar distribution in atoms among classes. Hence, this proposed approach can be applied to large amount of data. Experiments on standard datasets proved that the proposed approach not only retain discriminative information, but computationally efficient when compared to other similar kind of dictionary optimization. In the future work, we concentrate on updating representative dictionary atom of similar group with respect to removal of atoms in order to minimize loosing discriminative information.

CHAPTER 5

INFORMATION LOSS BASED SAMPLING FOR KERNEL DICTIONARY LEARNING

We remove redundant dictionary atoms to obtain compact and discriminative dictionary in the previous chapter. In this chapter, we incorporate kernelization of dictionary learning in an efficient way to obtain discriminative dictionary. Here, we propose an information loss based sampling to linearize kernel dictionary learning. Kernelization of K-SVD dictionary learning has been shown to achieve better classification performance than its linear counterpart. However, the process of kernelization generates kernel matrix and its dimension depends on total number of input samples. The size of kernel matrix increases when the number of input samples increases and this becomes computationally prohibitive. In order to solve this problem, the large kernel matrix has been approximated using well-known Nyström method in the literature. The Nyström method uses the subset of input samples for the approximation of large kernel matrix. So, the choice of sampling method results the goodness of the approximation of the kernel matrix. Hence, we introduce a sampling method based on information loss to approximate kernel matrix for the linearization of kernel dictionary learning. In this proposed sampling approach, computationally efficient *Jensen-Shannon divergence* is used to compare the probability distributions of input data given dictionary atom to merge similar dictionary atoms based on minimum information loss. This gives well discriminative samples which improves the kernel matrix approximation. We show the efficacy of the proposed sampling method through experimental results.

The non-linear mapping of input data into higher dimension has been well known to improve discriminability especially in classification. In the field of machine learning, this mapping generally referred to as functions called *kernels* and the mapping process popularly known as *kernelization.* The new space of this mapped signals in higher dimension is called *feature space.* This non-linear mapping from finite dimension to higher dimension can even be infinite which prohibits the learning process of classifier using signals in feature space. This issue can be tackled by *kernel trick* in which it computes inner product of the mapped signals without explicitly operating in its feature space. Kernel trick provides efficient computation of inner products of high dimensional vectors in the feature space and it can be applied to learning algorithms which fully posed in terms of inner product. In the process of *kernelization*, these inner products are replaced with kernels. The kernelization is successfully applied in many machine learning areas such as kernel-SVM [110] [111], kernel fisher discriminant [112] etc. and some of the popular kernels are linear, polynomial, Gaussian etc. The kernel matrix $\mathbf{K}$ is filled with the values from kernel function and the size of kernel matrix grows as number of input signals increases. The large number of input data results large kernel matrix which becomes serious issue while kernelizing the learning process. Here, this issue is addressed by approximating large kernel matrix in an efficient way.

The trend of kernelization is also ifluenced in the area of sparse representation and dictionary learning. Vincent and Bengio [113] kernelized the matching pursuit which looks for sparse kernel based solution for classification problems. Later, similar strategy is also applied to kernelize the basis pursuit algorithm by Guigue *et al.* [114]. The kernel sparse representations for machine learning applications such as visual tracking, face recognition, image classification are proposed by Wu [115] *et al.* Then, Gao *et al.* [116] used kernel sparse representation to project sparse coding technique into higher dimensional feature space, which is incorporated into spatial pyramid matching for image classification. In [117] [118], kernel sparse representation based classifier is applied on face database and authors reduced dimensionality of kernel feature space using a projection method. Harandi *et al.* [119] applied kernelization on the sparse coding algorithm LASSO for learning a Riemannian dictionary. As we have seen in the sparse coding, the kernelization has also been applied on dictionary learning. In [13]

[14] [15], Nguyen *et al.* propose an elegant approach to kernelize the K-SVD dictionary learning to obtain non-linear dictionary for object recognition and image classification. But large kernel matrix $\mathbf{K}$ is computationally prohibitive when using large dataset because size of the kernel matrix depends on number of training signals. Corts and Scott [120] suggested sparse approximation of kernel mean instead of involving all training signals. In this work, we concentrate on approximating large kernel matrix $\mathbf{K}$ to linearize kernel dictionary learning. In [12], Golts *et al.* approximate large kernel matrix $\mathbf{K}$ using Nyström method which is referred to as linearization of kernel dictionary learning (LKDL). The subset of input data (sampling) is to be obtained for Nyström approximation, so that a good sampling gives better approximation.

In this work, we propose a sampling technique based on *information loss* to improve Nyström approximation of the kernel matrix $\mathbf{K}$. This is inspired from *information bottleneck* approach [103] [102] in which mutual information loss is determined using distortion measure *Jensen-Shannon (JS) divergence* [17] which compare two probability distributions. Wilson and Mohan [121] use information bottleneck principle to obtain discriminative dictionaries for classification tasks. In [82], Tishby *et al.* analyze information loss at each layer in the deep neural network (DNN) based on information bottleneck principle, which helps to obtain optimal DNN for the given training data. In [78] [79], Krause *et al.* propose an optimal placement of sensors by maximizing mutual information based on Gaussian process (GP) which ultimately helps to reduce communication cost. Qiu *et al.* [2] maximize mutual information between (1) selected and unselected atoms, (2) sparse codes and class labels, (3) input signals and selected atoms, which result well representative dictionary atoms for image classification. But these works used *Gaussian Process* (GP) model in sparse representation which consumes much computational time to calculate inverse of the matrix. We compare dictionary atoms based on its sparse distribution over the input data to find information loss and merge similar dictionary atoms which are having minimum information loss. We use computationally efficient distortion measure JS divergence to determine the information loss. To compare performances, the proposed sampling technique is compared with other sampling techniques, viz., k-means, coresets, uniform, and diagonal sampling in experiments.

In this chapter, we discuss information loss based sampling for the linearization of kernel dictionary learning. In Section 5.1, classical dictionary learning approach and time complexity involved in the learning procedure. The kernelization of K-SVD dictionary is explained in the Section 5.2. The Section 5.3 describes the proposed approach of linearization of kernel dictionary learning. The experiments with standard datasets are conducted in the Section 5.4. Finally, the Section 5.5 summarizes the overall approach of linearizing kernel dictionary learning.

5.1 CLASSICAL DICTIONARY LEARNING

As we have discussed in previous chapters, classical dictionary learning is the state of art approach to learn directly from the input data, which can be attributed to better representation than predetermined dictionaries. There are many dictionary learning algorithms as discussed in the section 2.3. Here, we concentrate on K-SVD dictionary learning [61] which comprises two stages: OMP based *sparse coding* and SVD based *dictionary update*. The OMP uses l_0 norm to obtain sparse solution. Despite proved it's uniqueness and global optimality [83], there is no practical mechanism to obtain solution based on l_0 norm. In other words, this is an NP-hard problem. The OMP is a greedy approach to find l_0 norm solution. In OMP, nearly orthogonal dictionary atoms are selected to represent input vector $\mathbf{y}$ and $\mathbf{D}_{\mathcal{S}}$ contains the selected dictionary atoms. The set $\mathcal{S}$ consists of indices of selected dictionary atoms. So, the atom selection $(\mathbf{y} - \mathbf{D}_{\mathcal{S}}\mathbf{x}_{\mathcal{S}})^T\mathbf{d}_j \ \forall j \notin \mathcal{S}$ costs $O(m|\mathcal{S}| + m)$ and least squares to update solution, $\mathbf{x}_{\mathcal{S}} = (\mathbf{D}_{\mathcal{S}}^{\mathbf{T}}\mathbf{D}_{\mathcal{S}})^{-\mathbf{1}}\mathbf{D}_{\mathcal{S}}^{\mathbf{T}}\mathbf{y}$, costs $O(m|\mathcal{S}|^2 + m|\mathcal{S}| + |\mathcal{S}|^3)$. We will recall this observation in the section 5.2.

For dictionary update, each of the dictionary atom is to be updated sequentially using SVD as discussed in the section 2.3.3. To update k^{th} dictionary atom $\mathbf{d_k}$, error matrix $\mathbf{E_k}$ is obtained by removing $\mathbf{d_k}$ and corresponding sparse coefficients from

$\mathbf{Y} - \mathbf{DX}$, i.e. $\mathbf{E_k} = \mathbf{Y} - \sum_{\mathbf{j} \neq \mathbf{k}} \mathbf{d_j x^j}$, where $\mathbf{x^j}$ is j^{th} row of $\mathbf{X}$, that corresponds to dictionary atom $\mathbf{d_j}$. To retain sparsity, the input samples which are not used by atom $\mathbf{d_k}$ can be removed. For this purpose, zero coefficients are removed from $\mathbf{x^k}$ and denoted as $\mathbf{x^k_R}$. Then corresponding columns from $\mathbf{E_k}$ are to be removed and denoted as $\mathbf{E^R_k}$ which is to be decomposed by SVD to update dictionary atom $\mathbf{d_k}$. Next we discuss an approach to learn higher dimensional signals.

5.1.1 Double-sparsity model

To incorporate signals of large dimension, Rubinstein *et al.* [122] put forward the idea of sparse dictionary called double-sparsity model. This sparse structure fills the gap between *learning-based* dictionary and *analytic* dictionary which has efficient implementation but lacks adaptability like Wavelets [6], Curvelets [7] etc. The learning-based approach infers the dictionary from the set of training examples while analytic dictionaries are obtained from their algorithms. In the double-sparsity model, the dictionary $\mathbf{D} = \mathbf{\Theta A}$, where $\mathbf{\Theta}$ is *base dictionary* and $\mathbf{A}$ is *sparse dictionary.* This new structure can be included in the dictionary learning optimization task as follows:

$$\underset{\mathbf{A},\mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{\Theta A X}\|^{\mathbf{2}}_{\mathbf{F}} \quad \text{subject to} \quad \begin{cases} \forall j \ \ \|\mathbf{a_j}\|_{\mathbf{0}} = \mathbf{T_0}, \\ \forall i \ \ \|\mathbf{x_i}\|_{\mathbf{0}} \leq \mathbf{T_1}. \end{cases}$$

In this structure, dictionary atoms in $\mathbf{D}$ is described as linear combination of T_0 atoms over prespecified base dictionary $\mathbf{\Theta}$. The success of this model depends on the base dictionary $\mathbf{\Theta}$ which is to be computationally efficient. In [122], overcomplete discrete cosine transform is used as base dictionary while Sulam *et al.* [123] proposed cropped wavelet dictionary as base dictionary. The sparse matrix $\mathbf{X}$ can be obtained by any sparse coding algorithm with fixed $\mathbf{A}$. As we have seen in many dictionary learning algorithms, sequential update of atoms in the dictionary is performed on the following minimization form:

$$\underset{\mathbf{a_k}}{\operatorname{argmin}} \|\mathbf{E_k} - \mathbf{\Theta a_k x^k}\|^{\mathbf{2}}_{\mathbf{F}} \text{subject to} \quad \forall \mathbf{j} \ \ \|\mathbf{a_j}\|_{\mathbf{0}} = \mathbf{T_0},$$

where $\mathbf{E}_k = \mathbf{Y} - \sum_{j \neq k} \mathbf{\Theta} \mathbf{a}_j \mathbf{x}^j$ is the error matrix which is used to update the atom $\mathbf{a}_k$. Nguyen *et al.* [13] used this double sparsity model to kernelize dictionary learning.

5.2 KERNEL DICTIONARY LEARNING

Kernelization performs non-linear mapping of input data into higher dimensional space to improve discriminability in classification. Let $\Phi : \mathbb{R}^m \rightarrow \mathcal{F}$ be a function for non-linear mapping from m dimensional input signal to higher dimension called *feature space* $\mathcal{F}$. The kernel or kernel function, $\mathtt{k}$ is

$$\begin{aligned} \mathtt{k}(\mathbf{x}, \mathbf{y}) &=< \Phi(\mathbf{x}), \Phi(\mathbf{y}) > \\ &= \Phi(\mathbf{x})^T \Phi(\mathbf{y}), \end{aligned}$$

where $\Phi(\mathbf{x})$ and $\Phi(\mathbf{y})$ are mapped version of signals $\mathbf{x}$ and $\mathbf{y}$, respectively. The linear algorithm can be converted to non-linear by replacing its features with kernel function $\mathtt{k}(\cdot, \cdot)$. We have N input signals $\mathbf{Y} = [\mathbf{y}_1 \mathbf{y}_2 \dots \mathbf{y}_N] \in R^{m \times N}$ and the kernel matrix $\mathbf{K}$ is obtained by kernel values of pair of signals

$$\mathbf{K}_{ij} = \mathtt{k}(\mathbf{y}_i, \mathbf{y}_j) \qquad \forall\, i, j = 1 \dots N.$$

These kernel values can be obtained by kernel trick as discussed earlier. This kernel matrix, $\mathbf{K}$, is positive semi-definite (PSD) symmetric matrix, which satisfies Mercer's condition and generates a Reproducing Kernel Hilbert Space (RKHS).

To kernelize dictionary learning, the input signals and dictionary atoms are to be mapped into some feature space $\Phi(\mathbf{Y}) = [\Phi(\mathbf{y}_1)\Phi(\mathbf{y}_2) \dots \Phi(\mathbf{y}_N)]$ and $\Phi(\mathbf{D}) = [\Phi(\mathbf{d}_1)\Phi(\mathbf{d}_2) \dots \Phi(\mathbf{d}_K)]$, respectively, using mapping function Φ. Then the inner products in the learning algorithm can be replaced with kernel function $\mathbf{K}$. As in the double sparsity model, Nguyen *et al.* [13] used the multiplication of two dictionaries to form a structured dictionary. One dictionary is called *base dictionary* which contains mapped signals and another dictionary is *coefficient dictionary* whose atoms are updated during dictionary training. Here each dictionary atom lies within the subspace spanned by input signals, so the dictionary atoms in the feature space is written as linear combination of mapped input signals ie., $\Phi(\mathbf{D}) = \Phi(\mathbf{Y})\mathbf{A}$, where $\Phi(\mathbf{Y})$ is base dictionary and $\mathbf{A}$ is coefficient dictionary. The optimization problem becomes

$$\underset{\mathbf{A},\mathbf{X}}{\operatorname{argmin}} \|\Phi(\mathbf{Y}) - \Phi(\mathbf{Y})\mathbf{A}\mathbf{X}\|_F^2 \; s.t. \; \|\mathbf{x}_i\| \leq T \; \forall i = 1 \ldots N. \tag{5.1}$$

In this, $\Phi(\mathbf{Y})$ is fixed, only $\mathbf{A}$ will be updated during dictionary learning. Kernel dictionary learning has two stages like its linear counterpart, namely, sparse coding and dictionary update. We follow the idea given by Nguyen *et al.* [13] to kernelize K-SVD dictionary learning in which orthogonal matching pursuit (OMP) is used for sparse coding and dictionary update is carried out using singular value decomposition (SVD). In order to kernelize K-SVD dictionary learning, we need to kernelize OMP and SVD for sparse coding and dictionary update, respectively.

Kernel OMP: In this, we need to find sparse coefficients of dictionary atoms in feature space. The mapped signal $\Phi(\mathbf{y})$ of given signal $\mathbf{y} \in R^m$ can be approximated using few dictionary atoms in the feature space, i.e., $\Phi(\mathbf{y}) = \Phi(\mathbf{Y})\mathbf{A}_{\mathcal{S}}\mathbf{x}_{\mathcal{S}} + \mathbf{r}_{\mathcal{S}}$, where $\mathbf{A}_{\mathcal{S}}$ contains selected dictionary atoms and $\mathbf{x}_{\mathcal{S}}$ denotes corresponding coefficients. The set $\mathcal{S}$ consists of indices of selected dictionary atoms. The current residual $\mathbf{r}_{\mathcal{S}}$ is to be projected on remaining dictionary atoms as

$$\begin{aligned} \mathbf{r}_{\mathcal{S}}^T(\Phi(\mathbf{Y})\mathbf{a}_i) &= (\Phi(\mathbf{y}) - \Phi(\mathbf{Y})\mathbf{A}_{\mathcal{S}}\mathbf{x}_{\mathcal{S}})^T(\Phi(\mathbf{Y})\mathbf{a}_i) \\ &= (\mathbf{K}(\mathbf{y},\mathbf{Y}) - (\mathbf{A}_{\mathcal{S}}\mathbf{x}_{\mathcal{S}})^T\mathbf{K}(\mathbf{Y},\mathbf{Y}))\mathbf{a}_i, \end{aligned} \tag{5.2}$$

where kernel functions

$$\mathbf{K}(\mathbf{y},\mathbf{Y}) = [\mathtt{k}(\mathbf{y},\mathbf{y}_1)\mathtt{k}(\mathbf{y},\mathbf{y}_2)\ldots\mathtt{k}(\mathbf{y},\mathbf{y}_N)]$$

and

$$\begin{aligned} \mathbf{K}(\mathbf{Y},\mathbf{Y}) &= <\Phi(\mathbf{Y}),\Phi(\mathbf{Y})> \\ &= \mathtt{k}(\mathbf{y}_i,\mathbf{y}_j) \quad \forall i,j = 1\ldots N. \end{aligned}$$

Based on the largest projection, dictionary atom is selected from remaining dictionary atoms. To update entire $\mathbf{x}_{\mathcal{S}}$, the mapped signal $\Phi(\mathbf{y})$ is to be projected onto the subspace spanned by $\Phi(\mathbf{Y})\mathbf{A}_{\mathcal{S}}$. Then the updated coefficient vector $\mathbf{x}_{\mathcal{S}}$ becomes

$$\begin{aligned}\mathbf{x}_{\mathcal{S}} &= \Big(\big(\Phi(\mathbf{Y})\mathbf{A}_{\mathcal{S}}\big)^{T}\Phi(\mathbf{Y})\mathbf{A}_{\mathcal{S}}\Big)^{-1}\big(\Phi(\mathbf{Y})\mathbf{A}_{\mathcal{S}}\big)^{T}\Phi(\mathbf{y}) \qquad (5.3)\\ &= \big(\mathbf{A}_{\mathcal{S}}^{T}\mathbf{K}(\mathbf{Y},\mathbf{Y})\mathbf{A}_{\mathcal{S}}\big)^{-1}\big(\mathbf{K}(\mathbf{y},\mathbf{Y})\mathbf{A}_{\mathcal{S}}\big)^{T}.\end{aligned}$$

This procedure will be repeated until selection of T (sprsity constraint) dictionary atoms. Now this costs $O(N^2|\mathcal{S}|+N|\mathcal{S}|+|\mathcal{S}|^3)$ and computational complexity tremendously increased based on the number training samples N. We try to solve this problem in section 5.3 by linearizing kernel dictionary learning.

Kernel K-SVD: In this, sparse matrix $\mathbf{X}$ is fixed and coefficient matrix $\mathbf{A}$ to be updated. For the updation of dictionary $\mathbf{A}$, the optimization function $\|\Phi(\mathbf{Y})-\Phi(\mathbf{Y})\mathbf{A}\mathbf{X}\|_F^2$ is rewritten as

$$\|\Phi(\mathbf{Y})\Big(\mathbf{I}-\sum_{j\neq k}\mathbf{a}_j\mathbf{x}^j\Big)-\Phi(\mathbf{Y})(\mathbf{a}_k\mathbf{x}^k)\|_F^2, \qquad (5.4)$$

where $\mathbf{a}_k$ is the k^{th} column of coefficient matrix $\mathbf{A}$ and $\mathbf{x}^k$ is the k^{th} row of sparse matrix $\mathbf{X}$. Similar to the K-SVD dictionary learning, each of the dictionary atom is to be updated separately. As we can see in the equation (5.4), the dictionary atom $\mathbf{a}_k$ is to be updated by removing it from the error function $\mathbf{E}_k=\Big(\mathbf{I}-\sum_{j\neq k}\mathbf{a}_j\mathbf{x}^j\Big)$. To maintain sparsity, zero coefficients in $\mathbf{x}_k$ and its corresponding columns in $\mathbf{E}_k$ are to be removed. Then the optimization problem becomes

$$\|\Phi(\mathbf{Y})\mathbf{E}_k^R-\Phi(\mathbf{Y})(\mathbf{a}_k\mathbf{x}_R^k)\|_F^2,$$

where $\mathbf{E}_k^R$ and $\mathbf{x}_R^k$ denote $\mathbf{E}_k$ and $\mathbf{x}^k$ after removing unwanted columns, respectively. Now we can decompose $\Phi(\mathbf{Y})\mathbf{E}_k^R$ as rank-1 matrices using singular value decomposition (SVD), i.e.,

$$\Phi(\mathbf{Y})\mathbf{E}_k^R=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T, \qquad (5.5)$$

and then equate $\Phi(\mathbf{Y})\mathbf{a}_k\mathbf{x}_R^k$ with rank-1 matrix of largest singular value as

$$\Phi(\mathbf{Y})\mathbf{a}_k\mathbf{x}_R^k=\sigma_1\mathbf{u}_1\mathbf{v}_1^T, \qquad (5.6)$$

$$\mathbf{x}_R^k=\sigma_1\mathbf{v}_1^T, \qquad (5.7)$$

$$\Phi(\mathbf{Y})\mathbf{a}_k=\mathbf{u}_1, \qquad (5.8)$$

and singular values are arranged in descending order. So σ_1,$\mathbf{u}_1$, and $\mathbf{v}_1$ denote singular value at $\mathbf{\Sigma}(1,1)$, first column of matrix $\mathbf{U}$, and first column of matrix $\mathbf{V}$, respectively.

The direct decomposition of $\Phi(\mathbf{Y})\mathbf{E}_k^R$ is impractical because of high dimension of $\Phi(\mathbf{Y})$. This issue can be resolved by finding Gram matrix of $\Phi(\mathbf{Y})\mathbf{E}_k^R$ as

$$\begin{aligned}\big(\Phi(\mathbf{Y})\mathbf{E}_k^R\big)^T\big(\Phi(\mathbf{Y})\mathbf{E}_k^R\big) &= (\mathbf{E}_k^R)^T\mathbf{K}(\mathbf{Y},\mathbf{Y})(\mathbf{E}_k^R)\\ &= \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T.\end{aligned}$$

Then $\sigma_1 = \sqrt{\mathbf{\Sigma}^2(1,1)}$ and $\mathbf{v}_1$ is the first column of $\mathbf{V}$. Now multiply $\mathbf{V}$ on both sides of equation (5.5) and consider only first column, then

$$\Phi(\mathbf{Y})\mathbf{E}_k^R\mathbf{v}_1 = \sigma_1\mathbf{u}_1. \tag{5.9}$$

In equation (5.9), $\mathbf{u}_1$ can be substituted by equation (5.8) and we get $\Phi(\mathbf{Y})\mathbf{E}_k^R\mathbf{v}_1 = \sigma_1\Phi(\mathbf{Y})\mathbf{a}_k$. Then dictionary atom can be updated as

$$\mathbf{a}_k = \frac{1}{\sigma_1}\mathbf{E}_k^R\mathbf{v}_1. \tag{5.10}$$

This will be repeated for all K dictionary atoms.

5.3 LINEARIZED KERNEL DICTIONARY LEARNING

The major difficulty in kernelization is the handling of large Gram matrix or kernel matrix $\mathbf{K}$. The storage and computational complexity of kernel learning algorithm depends on the number of input samples N. The kernel matrix grows when the number of input samples increases. This becomes prohibitive in both storage and computational aspects. As we have seen in the kernel dictionary learning, large kernel matrix needs to be stored during sparse coding and dictionary update stage. This problem can be solved by approximating large kernel matrix by Nyström method without compromising classification accuracy. So, the kernel dictionary learning can be linearized by approximating large kernel matrix. In this low rank approximation, subset of the input samples are used and sampling the columns of input data matrix is very important to achieve good performance accuracy. In this work, we propose an efficient method to sample the kernel matrix which is discussed below.

5.3.1 Sampling based on information loss

Linearization of kernel dictionary learning using Nyström method uses subset of input data. To sample the input data, we propose a sampling method based on *information loss* among data. This is inspired from the classical information bottleneck approach proposed by Tishby *et al.* [103]. Initially, the input data $\mathbf{Y}_c$ of each class c is learned by K-SVD dictionary learning as

$$\underset{\mathbf{D}_c,\mathbf{X}_c}{\operatorname{argmin}} \|\mathbf{Y}_c - \mathbf{D}_c\mathbf{X}_c\|_F^2 \quad \text{subject to} \quad \forall i \ \ \|\mathbf{x}_i\|_0 \leq T, \tag{5.11}$$

where $\mathbf{D}_c$ and $\mathbf{X}_c$ are obtained dictionary and sparse matrix after learning, respectively. The sparse vector $\mathbf{x}_i \in \mathbf{X}_c$ corresponds to input vector $\mathbf{y}_i \in \mathbf{Y}_c$ and T is the sparsity constraint. Each dictionary atom lies within the subspace spanned by the input data and also there is a redundancy among the obtained dictionary atoms. By removing redundancy, we can obtain well representative dictionary atoms which can be used as a subset of input data for Nyström approximation. Now onwards, we denote $\mathbf{D}_c, \mathbf{X}_c, \mathbf{Y}_c$ as $\mathbf{D}, \mathbf{X}, \mathbf{Y}$, respectively, for the ease of use.

In this information theoretic approach, given empirical joint distribution of two random variables, we look for compact representation of one random variable which preserves as much as information about another random variable. More clearly, we compress the dictionary $\mathbf{D}$ into $\tilde{\mathbf{D}}$ which preserves maximum information about relevant variable $\mathbf{Y}$. Let $\mathbb{D}$, $\tilde{\mathbb{D}}$, and $\mathbb{Y}$ be random variable notation for $\mathbf{D}$, $\tilde{\mathbf{D}}$, and $\mathbf{Y}$, respectively. We denote probability mass function as $\mathsf{p(d)}$ and conditional distribution as $\mathsf{p(y|d)}$ rather than $\mathsf{p}_{\mathcal{D}}(\mathsf{d})$ and $\mathsf{p}_{\mathcal{Y}|\mathcal{D}}(\mathsf{y|d})$ for ease of use.

In this, we find compressed representation of $\mathbb{D}$, denoted by $\tilde{\mathbb{D}}$, such that mutual information $I(\tilde{\mathbb{D}};\mathbb{Y})$ is maximized while the constraint $I(\tilde{\mathbb{D}};\mathbb{D})$ is kept minimum. The mutual information is defined as

$$I(\mathbb{D};\mathbb{Y}) = \sum_{\mathsf{d}} \sum_{\mathsf{y}} \mathsf{p(d)p(y|d)} \log \frac{\mathsf{p(y|d)}}{\mathsf{p(y)}}, \tag{5.12}$$

and this gives the information measure that one random variable $\mathbb{D}$ contains about other random variable $\mathbb{Y}$. Our objective is to obtain compact representation $\tilde{\mathbb{D}}$ which retains maximum information about the relevant variable $\mathbb{Y}$, ie., maximize $I(\tilde{\mathbb{D}};\mathbb{Y})$. The compactness of the representation is determined by $I(\tilde{\mathbb{D}};\mathbb{D})$ which is to be minimized. Fortunately, this problem has an exact optimal solution without any assumption about the origin of the joint distribution $\mathsf{p(d,y)}$. The solutions to this problem are three probability distributions given as

$$\begin{cases} \mathsf{p(\tilde{d}|d)} = \frac{\mathsf{p(\tilde{d})}}{\mathcal{N}(\mathsf{d},\beta)}\mathsf{exp}\Big(-\beta\,\mathsf{D_{KL}}\big[\mathsf{p(y|d)}||\mathsf{p(y|\tilde{d})}\big]\Big) \\ \mathsf{p(y|\tilde{d})} = \frac{1}{\mathsf{p(\tilde{d})}}\sum_{\mathsf{d}}\mathsf{p(y|d)p(\tilde{d}|d)p(d)} \\ \mathsf{p(\tilde{d})} = \sum_{\mathsf{d}}\mathsf{p(\tilde{d}|d)p(d)}. \end{cases} \tag{5.13}$$

The details of proof are given in [103]. In general, the membership probabilities are *soft* because every $\mathsf{d} \in \mathrm{D}$ can be assigned to every $\tilde{\mathsf{d}} \in \tilde{\mathrm{D}}$ with a certain probability. As you can see in the equation (5.13), $\mathsf{p(\tilde{d}|d)}$ determines the distortion between two conditional probability distributions over the relevant variable $\mathbb{Y}$ using relative entropy or *Kullback-Leibler* divergence [16], ie., $\mathsf{D_{KL}(p||q)} = \sum_{\mathsf{x}}\mathsf{p(x)log}\frac{\mathsf{p(x)}}{\mathsf{q(x)}}$. Where $\mathcal{N}(\mathsf{d},\beta)$ is normalization factor and Lagrange multiplier β determines the *softness* of quantization. In this approach, the information contained in $\mathbb{D}$ about $\mathbb{Y}$ is squeezed through a compact *bottleneck* of dictionary atoms in $\tilde{\mathbb{D}}$. So, the compact representation keeps the relevant part (discriminative information) in $\mathbb{D}$ about the input $\mathbb{Y}$.

Here, we would prefer the simple implementation of information bottleneck approach called *Agglomerative information bottleneck* [102] in which it is restricted to *hard* partitions, ie., $\beta \to \infty$. In this case, each dictionary atom $\mathsf{d} \in \mathrm{D}$ belongs to only one of the partition $\tilde{\mathsf{d}} \in \tilde{\mathrm{D}}$. Then the probability distributions in the equation (5.13) becomes

$$\begin{cases} \mathsf{p(\tilde{d}|d)} = \begin{cases} 1 & \mathsf{d} \in \tilde{\mathsf{d}} \\ 0 & \text{otherwise} \end{cases} \\ \mathsf{p(\tilde{d})} = \sum_{\mathsf{d}\in\tilde{\mathsf{d}}}\mathsf{p(d)} \\ \mathsf{p(y|\tilde{d})} = \frac{1}{\mathsf{p(\tilde{d})}}\sum_{\mathsf{d}\in\tilde{\mathsf{d}}}\mathsf{p(y|d)p(d)}. \end{cases} \tag{5.14}$$

Now one can easily determine the mutual information $I(\tilde{\mathbb{D}};\mathbb{Y})$. So, dictionary atoms can be merged based on *information loss*, $\delta I_c = I(\tilde{\mathbb{D}}_{\text{before}};\mathbb{Y}) - I(\tilde{\mathbb{D}}_{\text{after}};\mathbb{Y})$, where $I(\tilde{\mathbb{D}}_{\text{before}};\mathbb{Y})$ and $I(\tilde{\mathbb{D}}_{\text{after}};\mathbb{Y})$ denote information measure before and after the merging process. The information loss is also rewritten [102] as

$$\delta I_c = \big(\mathsf{p}(\tilde{\mathsf{d}}_i) + \mathsf{p}(\tilde{\mathsf{d}}_j)\big)\mathsf{D}_{\mathsf{JS}}\big[\mathsf{p}(\mathsf{y}|\tilde{\mathsf{d}}_i), \mathsf{p}(\mathsf{y}|\tilde{\mathsf{d}}_j)\big], \tag{5.15}$$

where D_{JS} is *Jensen-Shannon (JS)* divergence [17] which is defined as

$$\mathsf{D}_{\mathsf{JS}}(\mathsf{p}_1, \mathsf{p}_2, \ldots, \mathsf{p}_n) = \mathsf{H}\Big(\sum_i \pi_i \mathsf{p}_i\Big) - \sum_i \pi_i \mathsf{H}(\mathsf{p}_i), \tag{5.16}$$

where $H(\mathbb{X}) = \sum_{\mathsf{x}} \mathsf{p}(\mathsf{x})\mathsf{log}\,\frac{1}{\mathsf{p}(\mathsf{x})}$ and $\mathsf{D}_{\mathsf{JS}} \geq 0$. However, if equality holds, distributions are identical. Here, the merging process takes place in a manner that two dictionary atoms are merged at a time to ensure optimal merge. The initial joint distribution, $\mathsf{p}(\mathsf{d}, \mathsf{y})$, is obtained from the sparse matrix $\mathbf{X}$ as

$$\mathsf{p}(\mathsf{d}, \mathsf{y}) = \frac{|\mathbf{X}(\mathsf{d}, \mathsf{y})|}{\sum_{\mathsf{d}} \sum_{\mathsf{y}} |\mathbf{X}(\mathbf{d}, \mathbf{y})|}. \tag{5.17}$$

Now we look two similar dictionary atoms to merge using information loss. Initially, $\tilde{\mathrm{D}} = \mathrm{D}$ and $\mathsf{p}(\tilde{\mathsf{d}}, \mathsf{y}) = \mathsf{p}(\mathsf{d}, \mathsf{y})$, ie., each dictionary atom is considered as compressed representation. One dictionary atom $\tilde{\mathsf{d}}$ is removed to form $\tilde{\mathrm{D}}^r = \Big\{\tilde{\mathrm{D}} - \{\tilde{\mathrm{d}}\}\Big\}$ and obtained joint distribution $\mathsf{p}(\tilde{\mathsf{d}}^r, \mathsf{y})$. Then we find most similar dictionary atom to $\tilde{\mathsf{d}}$ in $\tilde{\mathrm{D}}^r$ based on minimum information loss $\delta I_c\Big(\mathsf{p}(\mathsf{y}|\tilde{\mathsf{d}}), \mathsf{p}(\mathsf{y}|\tilde{\mathsf{d}}^r)\Big)$, $\forall \tilde{\mathsf{d}}^r \in \tilde{\mathrm{D}}^{\mathrm{r}}$. In this way, we can find similar atom of every dictionary atoms $\tilde{\mathsf{d}} \in \tilde{\mathrm{D}}$. Suppose, if similar atom of $\tilde{\mathsf{d}}_i$ is $\tilde{\mathsf{d}}_j$ and similar atom of $\tilde{\mathsf{d}}_j$ is $\tilde{\mathsf{d}}_i$, then these two can be merged as $< \tilde{\mathsf{d}}_i, \tilde{\mathsf{d}}_j > \rightarrow \tilde{\mathsf{d}}_*$. After the merging process, the probability distributions in equation (5.14) are to be updated as

$$\begin{cases} \mathsf{p}(\tilde{\mathsf{d}}_*|\mathsf{d}) = \begin{cases} 1 & \mathsf{d} \in \{\tilde{\mathsf{d}}_i, \tilde{\mathsf{d}}_j\} \\ 0 & \text{otherwise} \end{cases} \\ \mathsf{p}(\tilde{\mathsf{d}}_*) = \mathsf{p}(\tilde{\mathsf{d}}_i) + \mathsf{p}(\tilde{\mathsf{d}}_j) \\ \mathsf{p}(\mathsf{y}|\tilde{\mathsf{d}}_*) = \frac{1}{\mathsf{p}(\tilde{\mathsf{d}}_*)}\Big(\mathsf{p}(\tilde{\mathsf{d}}_i, \mathsf{y}) + \mathsf{p}(\tilde{\mathsf{d}}_j, \mathsf{y})\Big) \end{cases} \tag{5.18}$$

The above procedure can be repeated until no similar atoms found. Now we can use this proposed sampling approach for Nyström approximation as discussed in the next subsection.

5.3.2 Nyström method for approximation

The approximation of large matrix is inevitable in kernel based problems. An efficient approximation can be achieved by well known Nyström method [124] in which subset of input data has been used. The chosen subset is important for good approximation of large matrix. In the pioneer work, Williams and Seeger [124] used uniform sampling without replacement. Here, we propose an efficient sampling technique using information bottleneck in the section 5.3.1. We need to approximate large kernel matrix $\mathbf{K} \in R^{N\times N}$ which is positive semi definite (PSD) symmetric matrix. There are c samples, $\mathbf{C} \in R^{m\times c}$, obtained using information bottleneck method. Let $\mathbf{K}_{Nc} = \Phi(\mathbf{Y})^T\Phi(\mathbf{C})$ and $\mathbf{K}_{cc} = \Phi(\mathbf{C})^T\Phi(\mathbf{C})$ are kernel values obtained using subset $\mathbf{C}$ and input data $\mathbf{Y}$. The Nyström method approximates kernel matrix $\mathbf{K}$ using $\mathbf{K}_{Nc}$ and $\mathbf{K}_{cc}$, i.e.,

$$\mathbf{K} \approx \mathbf{K}_{Nc}\mathbf{K}_{cc}^{-1}\mathbf{K}_{Nc}^T. \tag{5.19}$$

The symmetric matrix $\mathbf{K}_{cc}$ can be eigen decomposed as $\mathbf{K}_{cc} = \mathbf{V\Sigma V^T}$, where $\mathbf{\Sigma}$ is diagonal matrix which contains eigen values in descending order and $\mathbf{V}$ denotes corresponding orthonormal eigen vectors. Then the equation (5.19) can be written as

$$\begin{aligned}\mathbf{K}_{Nc}\mathbf{K}_{cc}^{-1}\mathbf{K}_{Nc}^T &= \mathbf{K}_{Nc}(\mathbf{V\Sigma V}^T)^{-1}\mathbf{K}_{Nc}^T \\ &= \mathbf{K}_{Nc}\mathbf{V\Sigma^{-1}V}^T\mathbf{K}_{Nc}^T,\end{aligned} \tag{5.20}$$

where the kernel matrix $\mathbf{K} = \Phi(\mathbf{Y})^T\Phi(\mathbf{Y})$. So, from the equation (5.20), the virtual samples $\Phi(\mathbf{Y})$ can be approximated as c dimensional feature vectors, ie.,

$$\Phi(\mathbf{Y})_c = \left(\mathbf{\Sigma}^{-1}\right)^{\frac{1}{2}}\mathbf{V}^T\mathbf{K}_{Nc}^T. \tag{5.21}$$

This leads the complexity of dictionary learning from $O(N^2)$ to $O(Nc)$. In fact, if the number of input signals, N, is very large, then c can be tremendously reduced, i.e., $c \ll N$. The dimension of virtual samples can even be reduced by selecting p, ($p \leq c$), largest eigen values from equation (5.21)

$$\Phi(\mathbf{Y})_p = \left(\mathbf{\Sigma}^{-1}\right)_p^{\frac{1}{2}}\mathbf{V}_p^T\mathbf{K}_{Nc}^T. \tag{5.22}$$

These obtained kernelized features, $\Phi(\mathbf{Y})_p$, are referred as *virtual samples* which can be applied to any off-the-shelf dictionary learning such as K-SVD, LC-KSVD, FDDL etc. The virtual samples also can be obtained from the test data using same samples in $\mathbf{C}$. In this case, $\mathbf{K}_{Nc}$ is obtained from test samples, ie., $\mathbf{K}_{Nc} = \Phi(\mathbf{Y}_{\text{test}})^T\Phi(\mathbf{C})$. We also directly used this virtual samples in KKSVD of Nguyen *et al.* [13] which explained in section 5.2. The vitual samples are finite dimension, so this will not prohibit learning as in the kernelized samples in *feature space.* Atom selection and solution updation can be calculated from equations (5.2) and (5.3), respectively. To update the dictionary atom, $\Phi(\mathbf{Y})$ can be applied to equation (5.5) as

$$\Phi(\mathbf{Y})_p\mathbf{E}_k^R = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{5.23}$$

Unlike in section 5.2, $\mathbf{v}_1$ can be directly obtained from equation (5.23). Then the dictionary atom $\mathbf{a}_k$ can be updated using equation (5.10). In this way, computational time can be reduced as compared to original KKSVD [13] dictionary learning.

5.4 EXPERIMENTAL RESULTS

We have conducted different experimental comparisons to show the efficacy of proposed linearized kernel dictionary learning (LKDL) using sampling based on *information loss.* The digit datasets USPS and MNIST are used for experiments with same parameters as in [12] for better comparison. In addition to these datasets, we also used challenging action dataset UCF 10 and HMDB 51 for the experiments in which parameters are estimated empirically. Each action video is represeted by action bank feature [95]. Different sampling techniques such as k-means, corset, diagonal, uniform are compared with proposed sampling based on *information loss.* In [124], Williams *et al.* used uniform sampling without replacement. Zhang *et al.* [125] used k-means for sampling and cluster centers become samples. Diagonal sampling is non-uniform sampling and weights are obtained from the diagonal elements. Another non-uniform sampling suggested by Feldman *et al.* [126] for dictionary learning known in computational engineering as coresets. Randomness plays an important role in coresets like uniform sampling, so the result is fluctuated when we use small number of training samples.

All datasets are normalized to unit norm and mean extracted. The Gaussian kernel with values $\sigma = [0.5, 1, 2]$ and polynomial kernel of degree [2,3,4] are considered in this experiment. The parameter for Nyström approximation c depends on the size of the dictionary obtained from the input data. The dimension of virtual samples, p, is determined by selecting all eigen values greater than 0.01 as shown in equation (5.22). And the dimension of the virtual samples is same as the dimension of input signal when linear DL compares with LKDL. In *JS divergence*, equal weightages are given to each conditional distribution, ie., $\pi_1 = \pi_2 = 0.5$, in all experiments.

In our experiment, we focus on the following four benefits of the proposed approach: 1) overall improvement in discriminability than existing LKDL [12] and KKSVD [13], 2) minimum approximation error when compared to other sampling techniques, 3) to achieve better computational time than KKSVD by incorporating virtual samples directly into linear DL, and 4) reduced computational cost of KKSVD by providing virtual samples in equation (5.23). We use tools for OMP and KSVD from OMP-Box v10 and KSVD-Box v13, respectively, in the toolbox[1] provided by Rubinstein *et al.* [127]. For LKDL[2] and KKSVD[3], we have used code given by Golts *et al.* [12] and Nguyen *et al.* [13], respectively, to compare with our proposed approach. Moreover, all experiments are conducted on the same machine. As described in [12], the obtained kernelized features can be used in any off-the-shelf dictionary learning such as K-SVD, LC-KSVD, FDDL etc.

[1] Found in `http://www.cs.technion.ac.il/~ronrubin/software.html`

[2] Found in `www.cs.technion.ac.il/~elad/Various/LKDL_Package.rar`

[3] Found in `http://www.umiacs.umd.edu/~hien/KKSVD.zip`

5.4.1 Evaluation on USPS digit dataset

This dataset of handwritten digits includes 10 classes of 0-9 digits. Training and testing set consists of 7291 and 2007 images of digits. The size of each image is 16 × 16 which comprises feature vector of 256 dimension. All results are taken as average of 5 iterations because random initialization of dictionaries cause slight fluctuations in performance. The obtained virtual samples can be learned by linear DL in which 300 dictionary atoms are learned from each classes for the classification. The sparsity constraint $T = 5$ is determined empirically. Samples are obtained by merging similar atoms using proposed information loss sampling method. In the figure 5.1, first two columns in each row consists of similar dictionary atoms determined by information loss method and third column contains merged representation of these similar atoms. Similarly 4th and 5th columns in each row consists of similar dictionary atoms and 6th column contains merged one. This shows how well our proposed information loss approach determines similar dictionary atoms to obtain good samples.

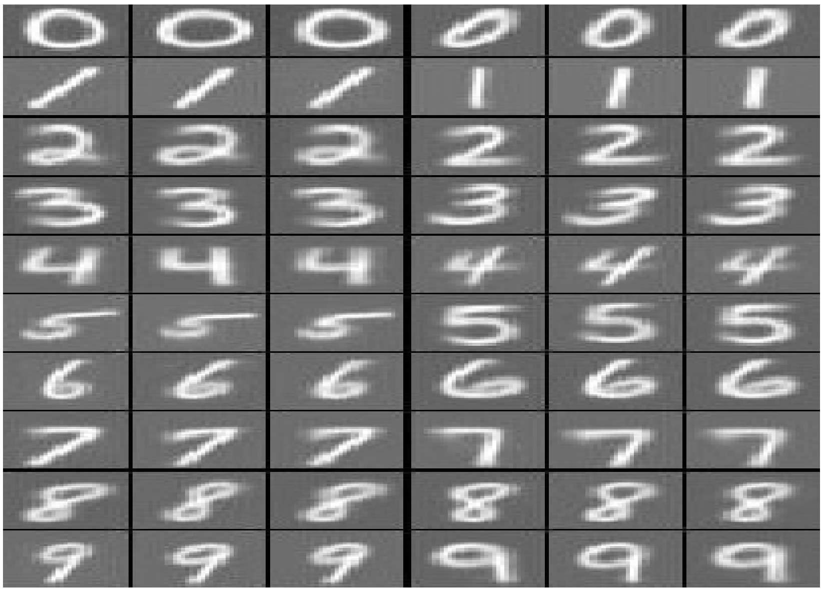


Fig. 5.1: Visualization of merge of similar dictionary atoms. merged atoms in 3rd and 6th column and similar atoms in its immediate prior columns

First, we compare the approximation error of proposed sampling approach with other methods. The approximation error is calculated as

$$\frac{\|\mathbf{K}-\tilde{\mathbf{K}}\|_{\mathbf{F}}}{\|\mathbf{K}\|_{\mathbf{F}}}. \tag{5.24}$$

The figure 5.2 shows sampling based on information loss gives minimum approximation error than other techniques. To corroborate the ability of our proposed sampling technique over other approaches, we compare recognition performance of different sampling techniques as shown in figure 5.3. In both cases, the proposed sampling method has clear advantage over others. Only k-means has good approximation error and recognition performance near to the proposed method, but k-means claims much computational time while dealing larger dataset which is discussed in the next experiment.

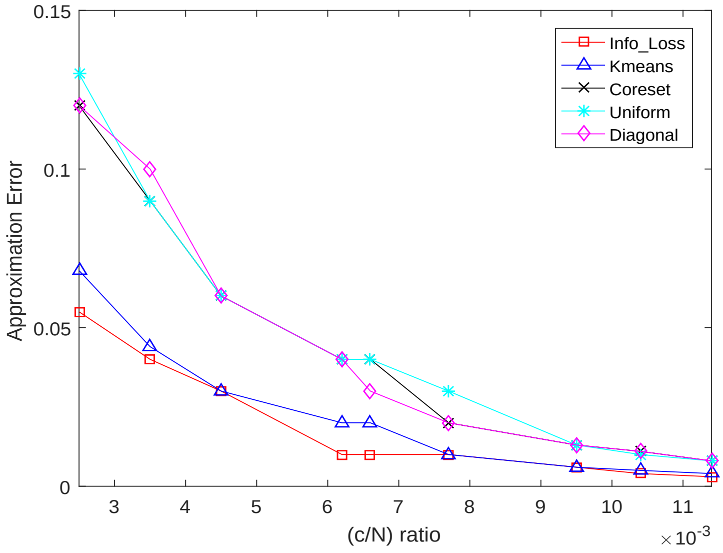


Fig. 5.2: Comparison of approximation error of proposed sampling approach with other techniques in USPS

5.4.2 Evaluation on MNIST dataset

MNIST also a digit dataset, but it's larger compared to USPS digit dataset. There are 60000 training samples and 10000 testing samples of digits of size 28 × 28. Each of the digits is arranged as vectors with dimension 784. Here, our aim is to measure computational efficiency of the proposed approach when it deals large dataset.

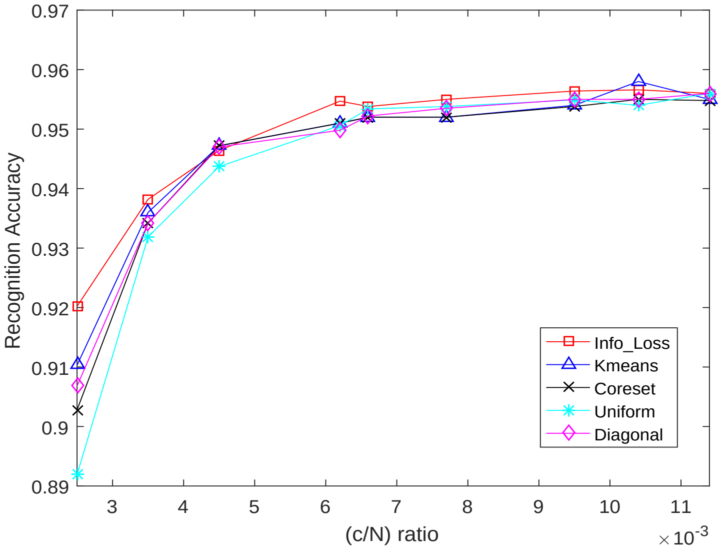


Fig. 5.3: Comparison of recognition accuracies of proposed sampling approach with other techniques in USPS

For training, 300 dictionary atoms are learned from each class of digits and sparsity constraint $T = 5$. The figure 5.4 compares recognition performances of different sampling techiniques. As expected, the k-means sampling is more close to the proposed approach, but former is computationally expensive when it deals larger dataset like MNIST as shown in figure 5.5. Next we compare linearized kernel dictionary (LKDL) using sampling based on information loss with kernel dictionary learning (KDL) and linear dictionary learning (DL) in the figure 5.6. It corroborates the kernelization of dictionary learning having clear advantage over the recognition accuracy, but it takes much computational time as shown in figure 5.7. Based on this experiment, we can see the linearization of kernel dictionary learning tremendously reduces the computational time.

5.4.3 Evaluation on UCF sports action data

This dataset contains 10 action videos of different sports such as diving, golfing, kicking, weight lifting, horse riding, running. skate boarding, swinging bench, swinging side angle and walking. Experiments are conducted such that 80% videos of each ac-

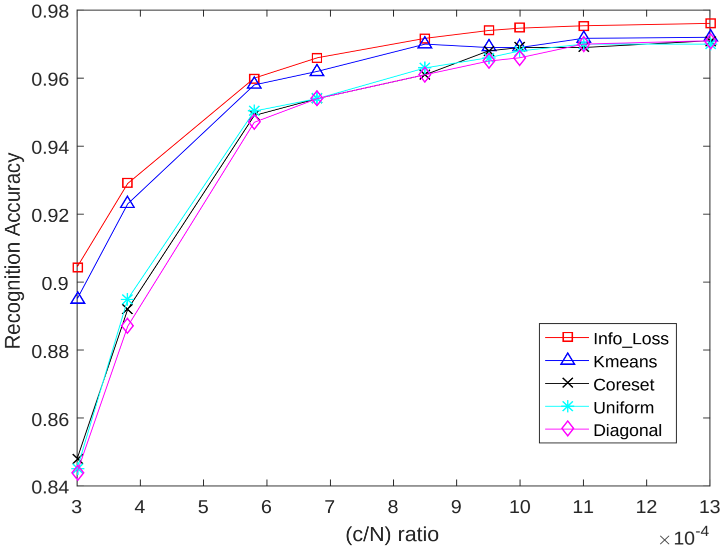


Fig. 5.4: Comparison of recognition accuracies of proposed sampling approach with other techniques in MNIST

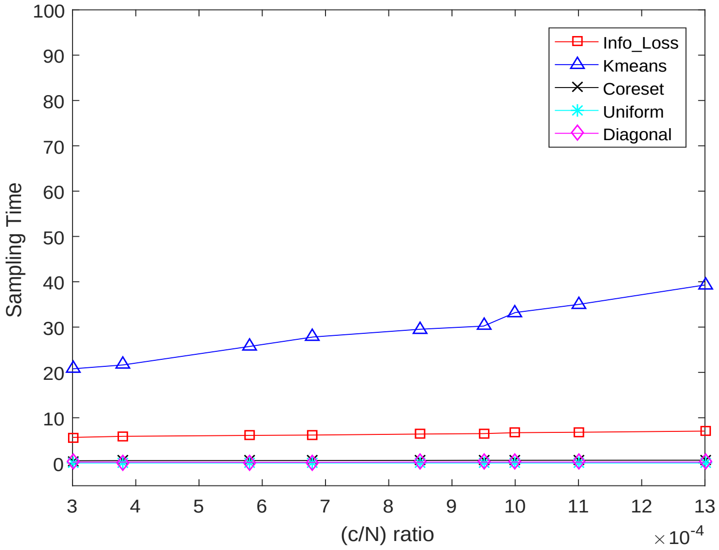


Fig. 5.5: Comparison of sampling time of proposed sampling approach with other techniques in MNIST

tion category randomly chosen as training data and remaining as testing data. The experiments are repeated 10 times and results are taken as average. For dictionary, 100 atoms are learned and sparsity constraint $T = 3$ is fixed empirically. We have

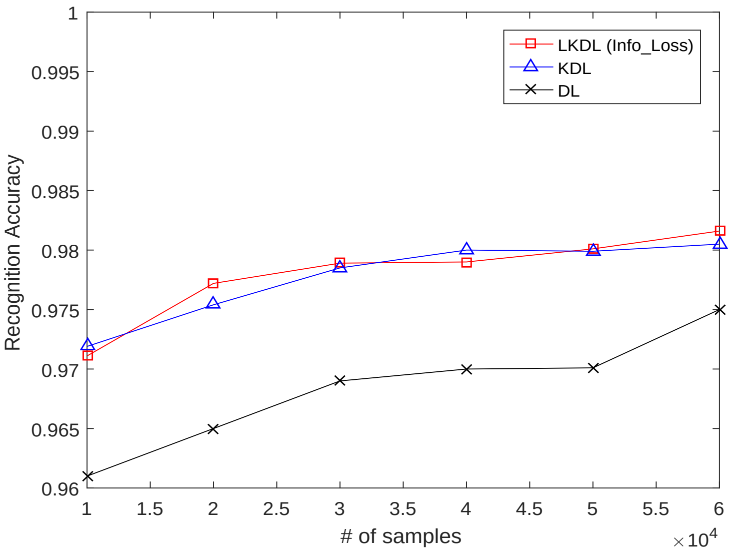


Fig. 5.6: Comparison of recognition accuracies of LKDL, KDL and DL in MNIST

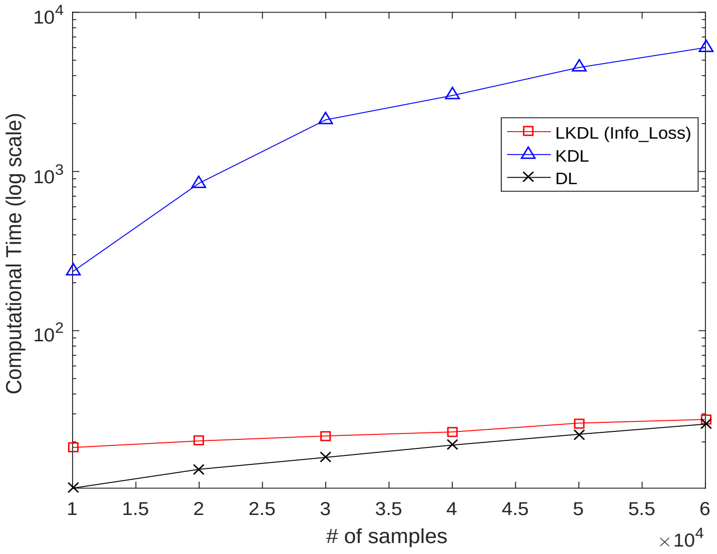


Fig. 5.7: Comparison of time taken for the computation of LKDL, KDL and DL in MNIST

conducted experiments with different sampling methods and compared recognition accuracy, sampling time, approximation error etc. as shown in the Table 5.1. Our proposed method achieved 98.10% accuracy which is better than performances in other methods. Wilson *et al.* [121], Sadanand *et al.* [95], Yao *et al.* [109], and Qiu *et al.*

[3] achieved 95.6%, 95%, 86.6%, and 83.6%, respectively. We obtained more than 2% improvement over the state of the art. As seen in the table 5.1, the proposed sampling approach achieved better recognition accuracy and approximation error than other methods which is clearly shows that our sampling technique provides better samples for the approximation. But our approach contribute little computational effort than other methods because size of training data is small in this dataset. The k-means sampling claims much computational time when it uses large training set. Our method computationally performs well even in large training data.

Table 5.1: Comparing performance of other sampling techniques with the proposed approach in UCF

	Rec. Acc. (%)	Samp. Time (sec.)	Appr. Error (sec.)
Info_Loss	**98.10**	**1.806**	**0.0124**
Kmeans	95.20	0.562	0.0318
Coreset	93.48	0.198	0.0498
Uniform	92.00	0.011	0.0214
Diagonal	92.60	0.029	0.0451

5.4.4 Evaluation on HMDB action data

The HMDB action dataset is a very challenging one. There are 51 classes of actions which are taken from various fields. To divide the input data into training and testing, we have followed the the same procedure as in UCF action dataset and results are taken as average of 10 iterations. From each action category, 50 dictionary atoms are learned and sparsity constraint T is set to 5. Table 5.2 compares sampling based on *information loss* with other similar techniques. In this, the proposed sampling gives better accuracy which is even better than the other approaches using the same action bank features. We achieved recognition accuracy of 35.39% compared to 26.9% [95] which is a benchmark result on HMDB dataset using action bank features. As you can see in the Table 5.2, we got better approximation error than other sampling techniques. In case of sampling time, the proposed method took 27.80 seconds compared to 60.86 seconds of kmeans which always tries to give closer accuracy to our approach. This proves the sampling based on information loss is a balanced approach with regards to recognition result and efficiency in computation.

Table 5.2: Comparing performance of other sampling techniques with the proposed approach in HMDB

	Rec. Acc. (%)	Samp. Time (sec.)	Appr. Error (sec.)
Info_Loss	**36.39**	**27.80**	**0.00062**
Kmeans	35.66	60.86	0.00062
Coreset	34.86	20.49	0.00092
Uniform	33.10	00.12	0.00088
Diagonal	34.10	01.58	0.00093

5.5 SUMMARY

In this chpater, we proposed an information loss based sampling to linearize kernel dictionary learning. This not only provides better sampling, it is also good in computational aspects. Because computing information loss using *Jensen Shannon divergence* is computationally efficient and it also good information theoretic measure to compare two probabilistic distribution. This is a balanced approach between recognition accuracy and computational time, so it improves overall recognition accuracy with minimum computational effort. Nyström method using obtained subsamples provides better approximation of large kernel matrix without compromising classification accuracy. The experimental results prove that this is an efficient approach to incorporate kernelization to obtain discriminative dictionary for classification tasks.

CHAPTER 6

COHERENT AND NONCOHERENT DICTIONARIES FOR CLASSIFICATION

In the previous chapters, we proposed different methods to obtain compact and discriminative dictionary using information theoretic approaches. The removal of redundant dictionary atoms and the incorporation of kernel features led to the compact and discriminative dictionary for classification. In this chapter, we propose an approach to obtain discriminative dictionary by exploiting underlying coherency among the input examples. First, the input data is divided into different clusters and the number of clusters depends on number of action categories. We seek data items of each action category within each cluster. If number of data items exceeds threshold in any action category, these items are labeled as *coherent*. In a similar way, all coherent data items from different clusters form a coherent group of each action category and data which are not part of the coherent group belong to *non-coherent* group of each action category. These coherent and non-coherent groups are separately learned using K-SVD dictionary learning. Since the coherent group has more similarity among data, only few atoms need to be learned. In non-coherent group, there is a high variability among the data items. So we propose an orthogonal projection based selection in non-coherent group to get optimal dictionary in order to retain maximum variance in the data. Finally, the obtained dictionary atoms of both groups in each action category are combined and then updated using Limited Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimization algorithm. The experiments are conducted on challenging datasets to validate efficacy of the proposed method.

In [128], input data is divided into clusters and learned into local dictionaries. The atoms of these local dictionaries are trained to obtain global dictionary. This helps to reduce computational time and increase performance in image processing applications. In our work, we treat coherent and non-coherent data items separately and learn them as separate dictionaries. Daniele *et al.* [129] learned dictionary with low mutual coherence by sparse representation and then update the dictionary using iterative projections and rotations. One of the main characterestics of dictionary learning is the mutual coherene among dictionary atoms. In order to reduce this mutual coherence, Mansour Nejati *et al.* [130] propose a coherence regularized dictionary learning which explicitly imposes a coherence regularizer while learning the dictionary. In [131], fixed coherence dictionary is made by maximizing pairwise decorrelations of atoms in the dictionary. The outline of the approach is shown in figure 6.1. In this work, we show how coherency among data can be exploited using the sparse based approach. For non-coherent data, an orthogonal projection based selection is used to obtain discriminative dictionary atoms. Then the obtained dictionary atoms are updated to enhance the recognition performance.

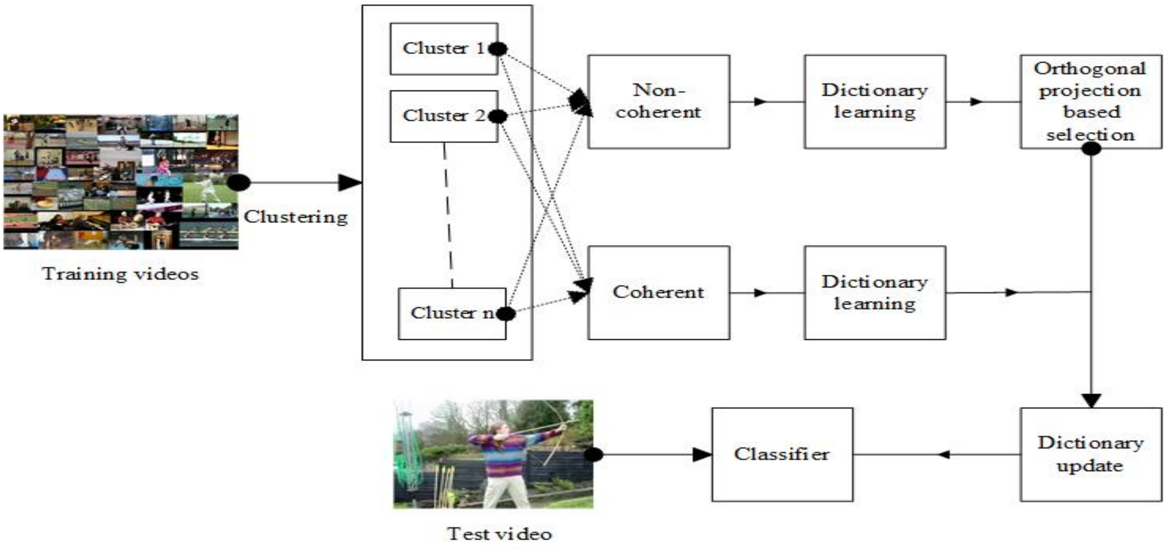


Fig. 6.1: Block diagram of the proposed approach. Dotted arrow denotes that cluster may or may not have coherent or noncoherent group.

In this chapter, we discuss the utilization of coherent and non-coherent input examples to obtain compact and discriminative dictionaries. The Section 6.1 describes coherent and non-coherent dictionary learning and then combine these two dictionaries for each action category. In the Section 6.2, the obtained dictionary is updated to get discriminability and the experiments using standard datasets are discussed in section 6.3. Finally, section 6.4 summarizes the entire proposed approach.

6.1 COHERENT AND NON-COHERENT DICTIONARY LEARNING

Initially, the input data $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N] \in R^{m \times N}$ is partitioned into n clusters using k-means and number of clusters, i.e. n, depends on number action categories in the dataset. We seek natural coherency by grouping input examples into n clusters. These clusters are $\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_n \quad \forall_i \, \mathbf{C}_i \in Y$ where $\mathbf{C}_i$ denotes i^{th} cluster. In each cluster, we look for coherent and non-coherent data items which are to be learned as separate dictionaries. Each of the coherent and non-coherent group is learned by K-SVD dictionary learning. As discussed in previous chapters, K-SVD performs sparse coding and dictionary update alternatively to find sparse matrix $\mathbf{X} \in R^{K \times N}$ and dictionary $\mathbf{D} \in R^{m \times K}$, respectively, in an iterative manner as

$$\underset{\mathbf{D},\mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{DX}\|_F^2 \quad \text{subject to} \quad \forall i \;\; \|\mathbf{x}_i\|_0 \leq T, \tag{6.1}$$

where the notation $\|.\|_F$ and $\|.\|_0$ denote frobenius norm and l_0 norm, respecively and each sparse vector $\mathbf{x_i} \in \mathbf{R^K}$ represents corresponding input vector $\mathbf{y}_i$. Sections 6.1.1 and 6.1.2 detail how to group and learn coherent and non-coherent data items.

6.1.1 Learning coherent actions

In each cluster $\mathbf{C}_i$, the data are grouped based on their action categories. For grouping, there is a constraint for minimum number of data items require to group. If it satisfies the constraint, then these grouped data items are labeled as *coherent*. Similarly, coherent data of particular action category, say c, are grouped from all clusters to form the coherent group $\mathbf{G}^c_{cohe}$ as

$$\mathbf{G}^c_{cohe} = [\mathbf{G}^c_1\mathbf{G}^c_2 \ldots \mathbf{G}^c_i \ldots \mathbf{G}^c_n], \quad 1 \leq c \leq p, i \in \{1, 2, \ldots, n\}$$

where p and n denote number of classes and number of clusters clusters having coherent data items, respectively. The coherent group, $\mathbf{G}^c_i$, may not exist in all clusters because of the minimum grouping constraint. Then each coherent group $\mathbf{G}^c_{cohe}$ is learned into the dictionary $\mathbf{D}^c_{cohe}$ using K-SVD dictionary learning. The coherent group contains similar data items, so that we can exploit sparsity by learning into few dictionary atoms. The advantage of this grouping is that only few dictionary atoms are required to approximate the input data which leads to the reduction of overall dictionary size and computational time. If there is more coherency in the input data, we can obtain very compact dictionary while achieving good recognition performance. All other data items which are not part of the coherent group belong to *non-coherent* group which is treated in a different manner as discussed in the next section.

6.1.2 Learning non-coherent actions

The non-coherent group has high variability among data items, because it is scattered in many clusters. So, we need to learn more dictionary atoms compared to coherent group discussed in the subsection 6.1.1. The selection of minimum number of discriminative dictionary atoms is a challenging task. As we did in the case of coherent group, non-coherent items in each action category c are grouped into $\mathbf{G}^c_{ncohe}$ and learned into the dictionary $\mathbf{D}^c_{cohe} = [\mathbf{d_1}\mathbf{d_2} \ldots \mathbf{d_k}]$, where $\mathbf{d}_i \in R^m$ represents dictionary atom. The goal is to select of most variant discriminative dictionary atoms from the dictionary. For this purpose, we propose orthogonal projection based selection to include maximum variability among the dictionary atoms for classification tasks. Here, one dictionary atom is to be picked randomly from $\mathbf{D}^c_{ncohe}$ and make it as residual vector $\mathbf{r}$. Now the current $\mathbf{D}^c_{ncohe}$ has only $(k-1)$ dictionary atoms. Initially, the closest dictionary atom from $\mathbf{D}^c_{ncohe}$ to the residual vector $\mathbf{r}$ to be found by projecting $\mathbf{r}$ onto the dictionary atoms. For this purpose, error $e(i)$ can be computed as

$$e(i) = min_{z_i} \|\mathbf{d}_i z_i - \mathbf{r}\|_2^2 \qquad \forall \mathbf{d}_i \in \mathbf{D}^c_{ncohe}, \tag{6.2}$$

and the optimal choice for z_i is

$$z_i^* = \frac{\mathbf{d}_i \cdot \mathbf{r}}{\|\mathbf{d}_i\|_2^2}, \tag{6.3}$$

where $\mathbf{d}_i \cdot \mathbf{r}$ denotes dot product between $\mathbf{d}_i$ and $\mathbf{r}$. Then, the closest vector $\mathbf{d}_{i_1}$ to $\mathbf{r}$ can be found by looking $e(i_1) \leq e(i)$ for all $\mathbf{d}_i$ in $\mathbf{D}^c_{ncohe}$. Then this $\mathbf{d}_{i_1}$ is removed from $\mathbf{D}^c_{ncohe}$ and added to empty set A. After getting $\mathbf{d}_{i_1}$, the residual $\mathbf{r}$ needs to be updated as $\mathbf{r} = \mathbf{r} - \mathbf{d}_{i_1} z^*_{i_1}$ and normalized to unit norm. The updated residual $\mathbf{r}$ is orthogonal to $\mathbf{d}_{i_1}$. In the next iteration, we can find $\mathbf{d}_i$ which is closest to the updated residual $\mathbf{r}$ using the same procedure. In each iteration, one vector from $\mathbf{D}^c_{ncohe}$ is chosen and added to set A . At the t^{th} iteration, A consist of t selected vectors viz. $\{\mathbf{d}_{i_1}, \mathbf{d}_{i_2}, \ldots, \mathbf{d}_{i_t}\}$ and then the updated residual becomes orthogonal to all dictionary atoms in A. So, the residual is updated as

$$\mathbf{r} = \mathbf{r} - \mathrm{A}(\mathrm{A}^T\mathrm{A})^{-1}\mathrm{A}^T\mathbf{r}, \tag{6.4}$$

where, with some abuse of notation, we use A to refer set of dictionary atoms as well as matrix of dictionary atoms. The set A usually contains only few atoms, so it does not take much computational time to calculate inverse of the matrix while updating residual.

The non-coherent dictionary after selecting most variant dictionary atoms denoted as $\mathbf{D}^{c^*}_{ncohe}$ which is cascaded to $\mathbf{D}^c_{cohe}$ to obtain final dictionary of action category c, ie., $\mathbf{D}^c = [\mathbf{D}^c_{cohe}\mathbf{D}^{c^*}_{ncohe}]$. Then the dictionary $\mathbf{D}^c$ to be updated which is discussed in the next section.

6.2 UPDATE THE DICTIONARY OF EACH ACTION

In each action category c, two dictionaries are obtained viz. $\mathbf{D}^c_{cohe}$ and $\mathbf{D}^{c^*}_{ncohe}$. These two dictionaries are cascaded to form dictionary $\mathbf{D}^c$ for each action category c. Here, we update the dictionary $\mathbf{D}^c$ using input data, $\mathbf{Y}^c$, of the action category c. An unconstrained non-linear optimization algorithm L-BFGS (Limited memory BFGS) [132] is used to update the dictionary. This approximates Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm using a limited amount of memory. It is based on gradient projection method. The matrix $\mathbf{Y}^c \in R^{m\times N^c}$ be the input data of action category c and N^c denotes number of input data belong to the same action category. The sparse matrix $\mathbf{X}^c \in R^{k\times N^c}$ can be obtained using OMP algorithm and this sparse matrix $\mathbf{X}^c$ is used to approximate the input data Y^c using dictionary D^c. Now the approximation becomes, $Y^c \approx D^c X^c$.

The cost function and gradient matrix are to be computed for the update. So the cost function J can be written as

$$J = \frac{1}{2N^c}\|\mathbf{D}^c\mathbf{X}^c - \mathbf{Y}^c\|_F^2 + \frac{\lambda}{2N^c}\sum_i\sum_j \mathbf{d}_{ij}^2, \tag{6.5}$$

where $\mathbf{d}_{ij}$ is the element in i^{th} row and j^{th} column in the matrix $\mathbf{D}^c$ and the regularization parameter λ is determined by empirically. The vectorized form of gradient matrix is formulated as

$$\frac{\partial J}{\partial \mathbf{D}^c} = \begin{bmatrix} \frac{\partial J}{\partial \mathbf{d}_{11}} & \frac{\partial J}{\partial \mathbf{d}_{12}} & \cdots & \frac{\partial J}{\partial \mathbf{d}_{1K}} \\ \frac{\partial J}{\partial \mathbf{d}_{21}} & \frac{\partial J}{\partial \mathbf{d}_{22}} & \cdots & \frac{\partial J}{\partial \mathbf{d}_{2K}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial J}{\partial \mathbf{d}_{n1}} & \frac{\partial J}{\partial \mathbf{d}_{n2}} & \cdots & \frac{\partial J}{\partial \mathbf{d}_{nK}} \end{bmatrix} = \frac{1}{N^c}(\mathbf{D}^c\mathbf{X}^c - \mathbf{Y}^c)\mathbf{X}^{cT} + \frac{\lambda}{N^c}\mathbf{D}^c. \tag{6.6}$$

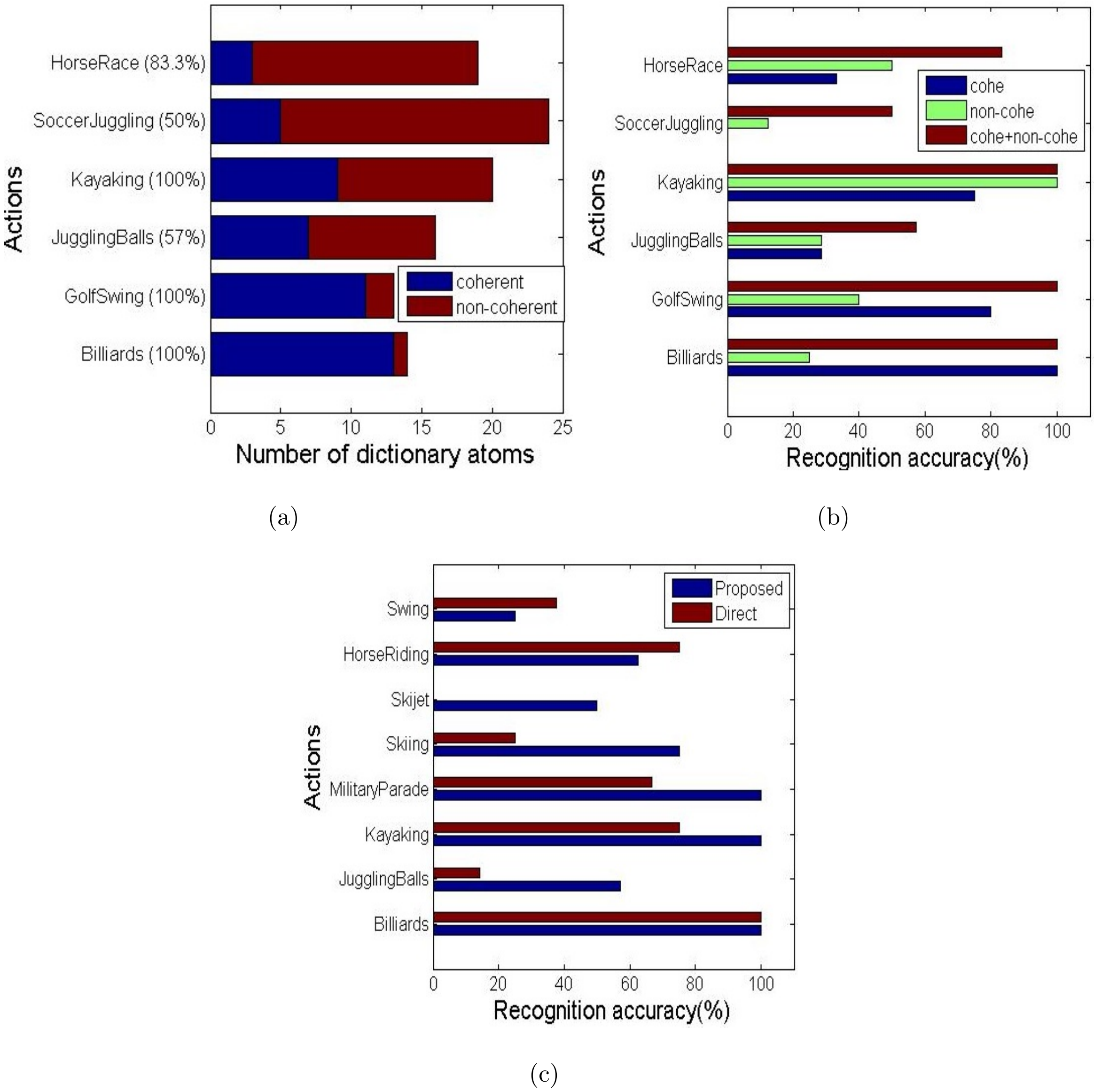
All updated dictionaries of each action categories are cascaded to form final dictionary $\mathbf{D} = [\mathbf{D}^1\mathbf{D}^2 \ldots \mathbf{D}^n]$. This dictionary $\mathbf{D}$ is used for the classification tasks.

6.3 EXPERIMENTAL RESULTS

We demonstrate our proposed approach on two challenging datasets, viz., UCF50 [133] and HMDB51 [98]. The Action bank [95] feature has been used to represent each action videos. Action bank comprises many individual action detectors, which constitutes mid-level representation of action data. The non-coherent groups are learned into dictionary of larger size as compared to coherent groups to maintain high variability in non-coherent group. However, in this experiment, coherent and non-coherent group are learned into dictionary size of 10% and 20% of input data, respectively. The sparsity constraint T is 10 and value of λ for dictionary update is 1. Moreover, the grouping constraint, ie., minimum number of coherent data items required to form the group, is taken as 10 in this experiment.

6.3.1 Evaluation on UCF50 action dataset

This is one of the challenging data set for action recognition. There are 50 action categories and 6950 action videos in all categories. There are 25 persons performing actions in each category. As the dataset consists of 50 classes, the input data is grouped into 50 clusters and each cluster is analysed for coherent and non-coherent data items. The obtained coherent and non-coherent dictionary are cascaded and updated as discussed in previous sections. The experimental results are taken based on Leave-One-Person-Out strategy. In figure 6.2(a), there are more number of coherent dictionary atoms than non-coherent in Golf swing and Billiards. In this case, it provides good recognition performance with least number of dictionary atoms which shows if coherency is more in any action category, then we can have better recognition while reducing overall dictionary size. Figure 6.2(b) shows recognition performance of coherent and non-coherent dictionary separately and both. It can be observed that both coherent and non-coherent dictionaries are contributing for the improvement of overall recognition accuracy. Then the proposed approach is compared with direct dictionary learning in the figure 6.2(c) which clearly indicates splitting the data into coherent and non-coherent is worth to enhance the recognition performance. The same number of atoms are used for both proposed and direct dictionary learning. Figure 6.3 depicts the performance of action recognition before and after the dictionary update. It can be seen that the dictionary update clearly enhances the overall recognition performance.



(a)

(b)

(c)

Fig. 6.2: The performance comparison: (a) no. of coherent and non-coherent dictionary atoms (b) coherent, non-coherent and combining both dictionary (c) proposed method and direct dictionary learning in UCF50.

6.3.2 Evaluation on HMDB51 action dataset

This is more challenging dataset compared to UCF50. It has 51 action categories and 6766 action videos. The input data are clustered into 51 clusters because dataset contains 51 classes. The results are obtained based on 10-fold cross validation. In this, most of the data items are grouped in non-coherent group as shown in figure 6.4(a),

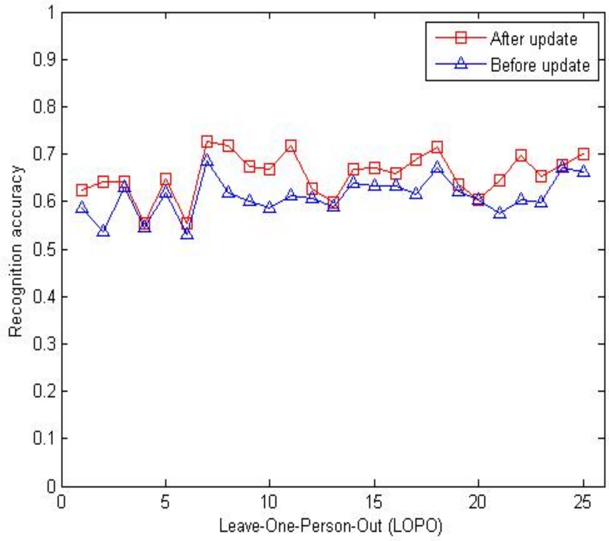


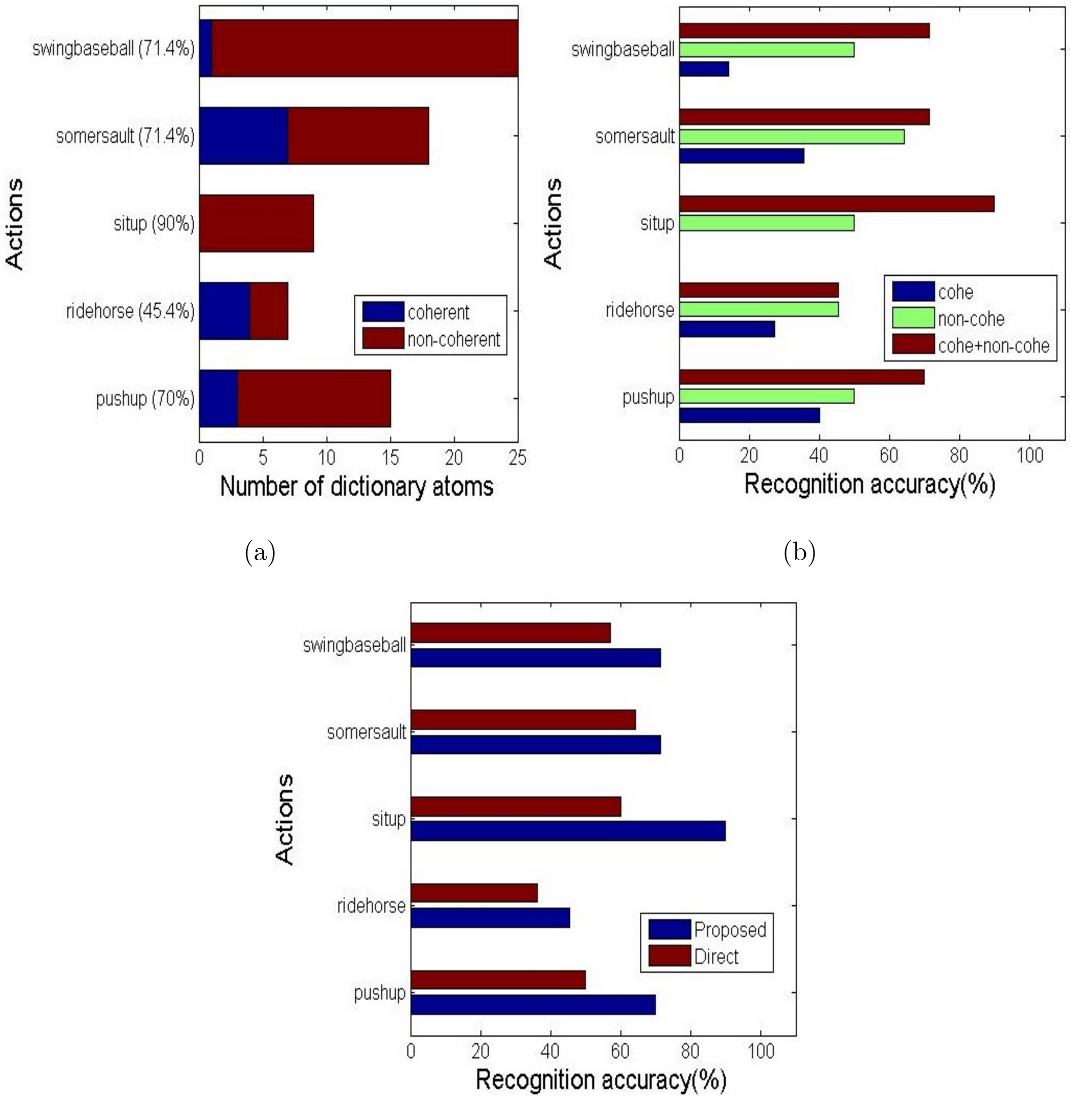
Fig. 6.3: Comparing performances of before and after dictionary update in UCF50. The x-axis indicates one of the 25 person taken as test data in LOPO evaluation.

this indicates the high variability in the dataset. As compared to coherent atoms, the non-coherent atoms are contributing more to the overall recognition performance as seen in figure 6.4(b). So the selection of non-coherent dictionary atoms is vital to this kind of challenging dataset. Figure 6.4(c) compares our proposed method with direct dictionary learning, which shows advantage of the proposed method by dividing input data into coherent and non-coherent.

6.3.3 Comparing with state of the art

In Table 6.1, we compare proposed method with other state of the art results in datasets UCF50 and HMDB51. Sadanand et al. [95] and shyju et al. [134] used same action bank features as ours and achieved performance of 57.9% and 59.3%, respectively. We improved this benchmark results using actionbank around 7%. Solmaz et al. [135] and Kliper el al. [136] achieved better performance than ours, but they used different features like GIST3D, MIP etc.

For HMDB51, our proposed method achieved better performance than all other



(a)

(b)

(c)

Fig. 6.4: The performance comparison: (a) no. of coherent and non-coherent dictionary atoms (b) coherent, non-coherent and combining both dictionary (c) proposed method and direct dictionary learning in HMDB51

state of art results. Sadanand et al. [95] got 26.9%, but we achieved remarkably good performance of 35.8% using action bank feature. Solmaz et al. [135] and Kliper el al. [136] obtained the performance of 29.2% and 29.17%, respectively. We could improve it further by around 6%.

Table 6.1: Comparing the proposed approach with the state of the art

Method	*Features*	*UCF50 (%)*	*HMDB51 (%)*
Sadanand et al. [95]	Action bank	57.90	26.90
Shyju et al. [134]	Action bank	59.30	23.62
Solmaz et al. [135]	GIST3D	73.70	29.20
Kliper-Gross et al. [136]	MIP	72.68	29.17
Proposed Method	Action bank	**66.30**	**35.8**

6.4 SUMMARY

We propose a novel approach to build compact and discriminative dictionaries by exploiting underlying coherency among the input examples in which the input data is divided coherent and non-coherent group and treated them separately. In this, the sparsity can be exploited among the coherent group which results in reduction of the size of the dictionary. If the input data has more coherent data, it can drastically reduce the overall dictionary size and computational time. In this way, the dictionary can be optimized effectively while keeping discriminative information for classification tasks. For non-coherent group, there is high variability among the data, so we use orthogonal projection based selection to get optimum discriminative dictionary atoms which is an efficient way to sustain high variability in the non-coherent data. This is a challenging task and we look more robust method in future work.

CHAPTER 7

SUMMARY AND CONCLUSIONS

In this thesis, new approaches were proposed to obtain compact and discriminative dictionaries for classification tasks. The dictionary learning is very powerful tool to represent signals, which provides an efficient way of adaptive learning from the input data. Though it is meant for signal reconstruction, this powerful tool can be used for classification tasks in an efficient manner. In order to obtain specific dictionary for classification purpose, there is a need to build discriminative dictionary in which atom possesses discriminative information with respect to classes. The issue of the standard dictionary learning is that it inherently produces redundant dictionary atoms for the purpose of reconstructing signals. But these redundant atoms are not significantly contributing to the discriminative nature of the dictionary. The ideal case is that the dictionary size should be minimum while keeping maximum discriminative dictionary atoms.

In this work, we propose an information bottleneck based approach to remove redundant dictionary atoms. This minimizes the mutual information between initial dictionary and optimized dictionary while maximizing the constraint of mutual information between optimized dictionary and class labels. This constraint information optimization provides self consistent equation which are used to determine the information loss between initial dictionary and optimized dictionary. The computation of information loss has been efficiently implemented using Jensen-Shannon divergence with adaptive weights. Based on the minimum loss of information, the redundant dictionary atoms are removed to obtain discriminative dictionary. This approach not only provides a naive way to build compact and discriminative dictionary especially for classification purpose, but also computationally efficient compared to other similar kind of state of the art approaches.

The kernelization is the traditional way to improve the discriminability in the field of machine learning. Here, we have addressed the issues related to the kernelization of dictionary learning to obtain discriminative dictionaries. The size of kernel matrix obtained through the process of kernelization depends on number of input examples which computationally prohibitive when the number of input examples increases. In literature, this large kernel matrix is approximated using well known Nyström method in which the input samples are taken for the approximation. The criteria used for sampling improves the Nyström approximation of the kernel matrix. In this thesis, we proposed an information loss based sampling for the Nyström approximation and experiments show that our approach performs well compared to other sampling methods. Unlike the previous approach, we remove one dictionary from initial dictionary and finds the similar dictionary atom, which is having similar sparse distribution over the input data, to the removed one. This approach slightly adds the computational effort compared to other random based sampling approaches, but proposed method helps to improve the approximation.

The other proposed approach is to obtain compact and discriminative dictionary based on the underlying coherency among the input examples. In this, the input data is divided as coherent and non-coherent group based on the coherence criteria. After obtaining coherent group, we exploited sparsity while learning dictionary atoms from the coherent group, which results in the reduction of dictionary size. In the non-coherent group, we tried to maximize discriminative atoms by projection technique. The learned dictionaries from both coherent and non-coherent are learned separately and cascaded to form single dictionary for particular action category. Finally, dictionaries from each action categories are updated to obtain discriminative dictionary.

7.1 CONTRIBUTIONS OF THE WORK

The important contributions of research work carried out as part of this thesis can be summarized as follows:

1. We utilize sparse distribution of atoms over the input examples to label and share dictionary atoms to find reconstruction error. This helps when all classes of input examples learned together.
2. A new information theoretic approach is proposed to optimize the dictionary which is suitable for classification tasks.
3. We combine dictionary learning and information bottleneck to obtain compact and discriminative dictionary. In this, the redundant dictionary atoms are removed in an efficient manner while keeping relevant information with respect to corresponding classes.
4. *Jensen-Shannon divergence* has been used to find similarities among class distribution given different dictionary atoms in which we proposed adaptive weights based on the distribution of dictionary atoms among classes, ie., atoms which have been used more times by input examples attract more weights.
5. To improve discriminability by kernelizing the dictionary learning, we proposed an information loss based sampling for the better approximation of the large kernel matrix using Nyström method. Thus we can efficiently adapt kernelized features to improve the discriminability of the dictionary.
6. We proposed an idea to find similar dictionary atoms such that one atom is removed and compared sparse distribution of remaining atoms with removed dictionary atom to determine similar sparse distribution. By looking at similar sparse distribution, we can find similar dictionary atoms effectively. In this way, we can remove redundant dictionary atoms while keeping discriminative atoms.
7. We propose an approach to obtain compact and discriminative dictionary based on the underlying coherency among the input examples. To achieve this, we

divide the input data into coherent and non-coherent group in which coherent group can be learned into few dictionary atoms compared to non-coherent. After combining coherent and non-coherent dictionaries, we update the dictionary for further improvement in classification.

7.2 DIRECTIONS FOR FURTHER RESEARCH

In this thesis, we have focused to obtain discriminative dictionaries. The digital dataset is used directly in the learning process whereas action bank are used to represent the action videos used in experiments. In the future work, we look for features or representations, which are having more representational ability, for action videos. The good representation of action videos definitely improves the overall performance of classification tasks.

In the proposed information loss sampling strategy, one dictionary atom is removed to find its similar sparse distribution. This adds slight computational cost because we need to remove all atoms in the similar manner. Here, we will look for an efficient approach to tackle the issue. One easy way is to assign task on multiple machines because of its similarity computations are independent. But we look for conceptual way to solve the problem of finding similar sparse distributions. Another important issue to be addressed is that the partition of coherent and non-coherent data. We need to look for an efficient method to partition the data such that better coherent group can be formed to compact discriminative dictionaries. We would like to apply this approach on another domain which is hyperspectral images in which dictionary can provide good representation.

REFERENCES

[1] H. Lobel, R. Vidal, and A. Soto, “Learning shared, discriminative, and compact representations for visual recognition,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, pp. 2218–2231, Nov 2015.

[2] Q. Qiu, V. Patel, and R. Chellappa, “Information-theoretic dictionary learning for image classification,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, pp. 2173–2184, Nov 2014.

[3] Q. Qiu, Z. Jiang, and R. Chellappa, “Sparse dictionary-based representation and recognition of action attributes,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 707–714, Nov 2011.

[4] L. Liu, L. Wang, and C. Shen, “A generalized probabilistic framework for compact codebook creation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 224–237, Feb 2016.

[5] S. Madeo and M. Bober, “Fast, compact, and discriminative: Evaluation of binary descriptors for mobile applications,” *IEEE Transactions on Multimedia*, vol. 19, pp. 221–235, Feb 2017.

[6] S. Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way.* Academic Press, 3rd ed., 2008.

[7] E. J. Cands and D. L. Donoho, “Recovering edges in ill-posed inverse problems: optimality of curvelet frames,” *Ann. Statist.*, vol. 30, pp. 784–842, 06 2002.

[8] M. N. Do and M. Vetterli, “The contourlet transform: an efficient directional multiresolution image representation,” *IEEE Transactions on Image Processing*, vol. 14, pp. 2091–2106, Dec 2005.

[9] E. L. Pennec and S. Mallat, “Sparse geometric image representations with bandelets,” *IEEE Transactions on Image Processing*, vol. 14, pp. 423–438, April 2005.

[10] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Transactions on Signal Processing*, vol. 41, pp. 3397–3415, Dec 1993.

[11] S. Chen and D. Donoho, “Basis pursuit,” in *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 41–44 vol.1, Oct 1994.

[12] A. Golts and M. Elad, “Linearized kernel dictionary learning,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, pp. 726–739, June 2016.

[13] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, “Kernel dictionary learning,” in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2021–2024, March 2012.

[14] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Design of non-linear kernel dictionaries for object recognition," *IEEE Transactions on Image Processing*, vol. 22, pp. 5123–5135, Dec 2013.

[15] A. Shrivastava, H. V. Nguyen, V. M. Patel, and R. Chellappa, "Design of non-linear discriminative dictionaries for image classification," in *Computer Vision – ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part I* (K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu, eds.), pp. 660–674, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[16] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, pp. 79–86, 03 1951.

[17] J. Lin, "Divergence measures based on the shannon entropy," *Information Theory, IEEE Transactions on*, vol. 37, pp. 145–151, Jan 1991.

[18] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.

[19] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, pp. 90–93, Jan 1974.

[20] S. Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way.* Academic Press, 3rd ed., 2008.

[21] J. B. Allen and L. R. Rabiner, "A unified approach to short-time fourier analysis and synthesis," *Proceedings of the IEEE*, vol. 65, pp. 1558–1564, Nov 1977.

[22] A. Janssen, "Gabor representation of generalized functions," *Journal of Mathematical Analysis and Applications*, vol. 83, no. 2, pp. 377 – 394, 1981.

[23] M. J. Bastiaans, "Gabor's expansion of a signal into gaussian elementary signals," *Proceedings of the IEEE*, vol. 68, pp. 538–539, April 1980.

[24] I. Daubechies, A. Grossmann, and Y. Meyer, "Painless nonorthogonal expansions," *Journal of Mathematical Physics*, vol. 27, pp. 1271–1283, 05 1986.

[25] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Transactions on Information Theory*, vol. 36, pp. 961–1005, Sep 1990.

[26] J. Wexler and S. Raz, "Discrete gabor expansions," *Signal Process.*, vol. 21, pp. 207–220, Oct. 1990.

[27] S. Qian and D. Chen, "Discrete gabor transform," *IEEE Transactions on Signal Processing*, vol. 41, pp. 2429–2438, Jul 1993.

[28] J. G. Daugman, "Daugman, j.g.: Two-dimensional spectral analysis of cortical receptive field profile," *Vision Research*, vol. 20, pp. 847–856, 02 1980.

[29] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Am. A*, vol. 2, pp. 1160–1169, Jul 1985.

[30] J. G. Daugman, “Complete discrete 2-d gabor transforms by neural networks for image analysis and compression,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 1169–1179, Jul 1988.

[31] M. Porat and Y. Y. Zeevi, “The generalized gabor scheme of image representation in biological and machine vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 452–468, Jul 1988.

[32] P. Burt and E. Adelson, “The laplacian pyramid as a compact image code,” *IEEE Transactions on Communications*, vol. 31, pp. 532–540, Apr 1983.

[33] A. Grossmann and J. Morlet, “Decomposition of hardy functions into square integrable wavelets of constant shape,” *SIAM Journal on Mathematical Analysis*, vol. 15, no. 4, pp. 723–736, 1984.

[34] S. G. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, Jul 1989.

[35] S. Mallat and W. L. Hwang, “Singularity detection and processing with wavelets,” *IEEE Transactions on Information Theory*, vol. 38, pp. 617–643, March 1992.

[36] S. Mallat and S. Zhong, “Characterization of signals from multiscale edges,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 710–732, Jul 1992.

[37] S. Mallat and W. L. Hwang, “Singularity detection and processing with wavelets,” *IEEE Transactions on Information Theory*, vol. 38, pp. 617–643, March 1992.

[38] R. R. Coifman, Y. Meyer, and V. Wickerhauser, “Wavelet analysis and signal processing,” in *IN WAVELETS AND THEIR APPLICATIONS*, pp. 153–178, 1992.

[39] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, “Shiftable multiscale transforms,” *IEEE Transactions on Information Theory*, vol. 38, pp. 587–607, March 1992.

[40] G. Beylkin, “On the representation of operators in bases of compactly supported wavelets,” *SIAM Journal on Numerical Analysis*, vol. 29, no. 6, pp. 1716–1740, 1992.

[41] G. P. Nason and B. W. Silverman, “The stationary wavelet transform and some statistical applications,” in *Wavelets and Statistics* (A. Antoniadis and G. Oppenheim, eds.), pp. 281–299, New York, NY: Springer New York, 1995.

[42] 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.

[43] E. Candes and D. Donoho, “Curvelets: A Surprisingly Effective Nonadaptive Representation of Objects with Edges,” 1999.

[44] E. Cands, L. Demanet, D. Donoho, and L. Ying, "Fast discrete curvelet transforms," *SIAM Journal on Multiscale Modeling and Simulation*, vol. 5, pp. 861–899, 09 2006.

[45] L. Ying, L. Demanet, and E. Candes, "3d discrete curvelet transform," *SPIE: Wavelets XI*, vol. 5914, pp. 351–361, 2005.

[46] M. N. Do and M. Vetterli, "Contourlets: a new directional multiresolution image representation," in *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers, 2002.*, vol. 1, pp. 497–501, Nov 2002.

[47] Y. Lu and M. N. Do, "A new contourlet transform with sharp frequency localization," in *2006 International Conference on Image Processing*, pp. 1629–1632, Oct 2006.

[48] R. Eslami and H. Radha, "Translation-invariant contourlet transform and its application to image denoising," *IEEE Transactions on Image Processing*, vol. 15, pp. 3362–3374, Nov 2006.

[49] A. L. D. Cunha, J. Zhou, and M. N. Do, "The nonsubsampled contourlet transform: Theory, design, and applications," *IEEE Transactions on Image Processing*, vol. 15, pp. 3089–3101, Oct 2006.

[50] G. Peyré and S. Mallat, "Surface compression with geometric bandelets," in *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, (New York, NY, USA), pp. 601–608, ACM, 2005.

[51] I. W. Selesnick, R. G. Baraniuk, and N. C. Kingsbury, "The dual-tree complex wavelet transform," *IEEE Signal Processing Magazine*, vol. 22, pp. 123–151, Nov 2005.

[52] G. Easley, D. Labate, and W.-Q. Lim, "Sparse directional image representations using the discrete shearlet transform," *Applied and Computational Harmonic Analysis*, vol. 25, no. 1, pp. 25 – 46, 2008.

[53] V. Velisavljevic, B. Beferull-Lozano, M. Vetterli, and P. L. Dragotti, "Directionlets: anisotropic multidirectional representation with separable filtering," *IEEE Transactions on Image Processing*, vol. 15, pp. 1916–1933, July 2006.

[54] S. Mallat, "Geometrical grouplets," *Applied and Computational Harmonic Analysis*, vol. 26, no. 2, pp. 161 – 180, 2009.

[55] 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.

[56] B. A. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 06 1996.

[57] K. Engan, S. O. Aase, and J. H. Husoy, "Method of optimal directions for frame design," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 5, 1999.

[58] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, (New York, NY, USA), pp. 689–696, ACM, 2009.

[59] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, pp. 407–499, 2004.

[60] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[61] 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, pp. 4311–4322, Nov 2006.

[62] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pp. 40–44 vol.1, Nov 1993.

[63] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *2011 International Conference on Computer Vision*, pp. 543–550, Nov 2011.

[64] 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, pp. 210–227, Feb 2009.

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

[66] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised dictionary learning," in *Advances in Neural Information Processing Systems 21* (D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds.), pp. 1033–1040, Curran Associates, Inc., 2009.

[67] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 791–804, April 2012.

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

[69] Q. Zhang and B. Li, "Discriminative k-svd for dictionary learning in face recognition," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2691–2698, June 2010.

[70] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Metaface learning for sparse representation based face recognition," in *2010 IEEE International Conference on Image Processing*, pp. 1601–1604, Sept 2010.

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

[72] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2Nd Edition).* Wiley-Interscience, 2000.

[73] S. Cai, W. Zuo, L. Zhang, X. Feng, and P. Wang, "Support vector guided dictionary learning," in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), vol. 8692 of *Lecture Notes in Computer Science*, pp. 624–639, Springer, 2014.

[74] C. Deng, X. Tang, J. Yan, W. Liu, and X. Gao, "Discriminative dictionary learning with common label alignment for cross-modal retrieval," *IEEE Transactions on Multimedia*, vol. 18, pp. 208–218, Feb 2016.

[75] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent k-svd," in *CVPR 2011*, pp. 1697–1704, June 2011.

[76] Z. Jiang, Z. Lin, and L. Davis, "Label consistent k-svd: Learning a discriminative dictionary for recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, pp. 2651–2664, Nov 2013.

[77] S. Y. Lee, J. Y. Sim, C. S. Kim, and S. U. Lee, "Correspondence matching of multi-view video sequences using mutual information based similarity measure," *IEEE Transactions on Multimedia*, vol. 15, pp. 1719–1731, Dec 2013.

[78] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, "Near-optimal sensor placements: maximizing information while minimizing communication cost," in *Information Processing in Sensor Networks, 2006. IPSN 2006. The Fifth International Conference on*, pp. 2–10, 2006.

[79] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, pp. 235–284, June 2008.

[80] J. Liu and M. Shah, "Learning human actions via information maximization," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, June 2008.

[81] S. Lazebnik and M. Raginsky, "Supervised learning of quantizer codebooks by information loss minimization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, pp. 1294–1309, July 2009.

[82] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Information Theory Workshop (ITW), 2015 IEEE*, pp. 1–5, April 2015.

[83] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing.* Springer Publishing Company, Incorporated, 1st ed., 2010.

[84] 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.

[85] J. J. Hull, “A database for handwritten text recognition research,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 550–554, May 1994.

[86] L. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, pp. 257–286, Feb 1989.

[87] J. Kruskall and M. Liberman, “The symmetric time warping algorithm: From continuous to discrete,” *Time Warps*, pp. 125–162, 1983.

[88] A. Stefan, V. Athitsos, and G. Das, “The move-split-merge metric for time series,” *IEEE Trans. on Knowl. and Data Eng.*, vol. 25, pp. 1425–1438, June 2013.

[89] Y. G. Jiang, Z. Wu, J. Wang, X. Xue, and S. F. Chang, “Exploiting feature and class relationships in video categorization with regularized deep neural networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[90] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, June 2014.

[91] L. Zhang, T. Wang, and X. Zhen, “Recognizing actions via sparse coding on structure projection,” in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pp. 2412–2415, Sept 2013.

[92] C. Wang and H. Liu, “Unusual events detection based on multi-dictionary sparse representation using kinect,” in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pp. 2968–2972, Sept 2013.

[93] I. Jargalsaikhan, S. Little, C. Direkoglu, and N. O’Connor, “Action recognition based on sparse motion trajectories,” in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pp. 3982–3985, Sept 2013.

[94] H. Wang, C. Yuan, W. Hu, H. Ling, W. Yang, and C. Sun, “Action recognition using nonnegative action component representation and sparse basis selection,” *IEEE Transactions on Image Processing*, vol. 23, pp. 570–581, Feb 2014.

[95] S. Sadanand and J. J. Corso, “Action bank: A high-level representation of activity in video,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1234–1241, June 2012.

[96] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: A local svm approach,” in *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR’04) Volume 3 - Volume 03*, ICPR ’04, (Washington, DC, USA), pp. 32–36, IEEE Computer Society, 2004.

[97] M. D. Rodriguez, J. Ahmed, and M. Shah, “Action mach a spatio-temporal maximum average correlation height filter for action recognition,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, June 2008.

[98] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “Hmdb: A large video database for human motion recognition,” in *2011 International Conference on Computer Vision*, pp. 2556–2563, Nov 2011.

[99] T. Chen, K. H. Yap, and D. Zhang, “Discriminative soft bag-of-visual phrase for mobile landmark recognition,” *IEEE Transactions on Multimedia*, vol. 16, pp. 612–622, April 2014.

[100] T. Chen and K. H. Yap, “Context-aware discriminative vocabulary learning for mobile landmark recognition,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, pp. 1611–1621, Sept 2013.

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

[102] N. Slonim and N. Tishby, “Agglomerative information bottleneck,” in *Advances in Neural Information Processing systems (NIPS-12)*, pp. 617–623, MIT Press, 1999.

[103] N. Tishby, F. C. Pereira, and W. Bialek, “The information bottleneck method,” in *Communication, Control, and Computing, The 37’th Allerton Conference on*, pp. 368–377, 1999.

[104] T. M. Cover and J. A. Thomas, *Elements of Information Theory, 2nd edition.* John Wiley & Sons, Inc., New Jersey, 2006.

[105] R. Blahut, “Computation of channel capacity and rate-distortion functions,” *Information Theory, IEEE Transactions on*, vol. 18, pp. 460–473, July 1972.

[106] A. Martinez and R. Benavente, “The AR face database,” *CVC Technical Report*, June 1998.

[107] L. Yeffet and L. Wolf, “Local trinary patterns for human action recognition,” in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 492–497, Sept 2009.

[108] M. Rodriguez, J. Ahmed, and M. Shah, “Action mach a spatio-temporal maximum average correlation height filter for action recognition,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, June 2008.

[109] A. Yao, J. Gall, and L. Van Gool, “A hough transform-based voting framework for action recognition,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2061–2068, June 2010.

[110] V. N. Vapnik, *The Nature of Statistical Learning Theory.* New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[111] M. Hu, Y. Chen, and J. T. Y. Kwok, “Building sparse multiple-kernel svm classifiers,” *IEEE Transactions on Neural Networks*, vol. 20, pp. 827–839, May 2009.

[112] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers, “Fisher discriminant analysis with kernels,” in *Neural Networks for Signal Processing IX: Proceedings of the*

1999 IEEE Signal Processing Society Workshop (Cat. No.98TH8468), pp. 41–48, Aug 1999.

[113] P. Vincent and Y. Bengio, "Kernel matching pursuit," *Machine Learning*, vol. 48, no. 1, pp. 165–187, 2002.

[114] V. Guigue, A. Rakotomamonjy, and S. Canu, *Kernel Basis Pursuit*, pp. 146–157. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[115] Y. Wu, Y. Jia, P. Li, J. Zhang, and J. Yuan, "Manifold kernel sparse representation of symmetric positive-definite matrices and its applications," *IEEE Transactions on Image Processing*, vol. 24, pp. 3729–3741, Nov 2015.

[116] S. Gao, I. W.-H. Tsang, and L.-T. Chia, *Kernel Sparse Representation for Image Classification and Face Recognition*, pp. 1–14. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[117] L. Zhang, W. D. Zhou, P. C. Chang, J. Liu, Z. Yan, T. Wang, and F. Z. Li, "Kernel sparse representation-based classifier," *IEEE Transactions on Signal Processing*, vol. 60, pp. 1684–1695, April 2012.

[118] M. Jian and C. Jung, "Class-discriminative kernel sparse representation-based classification using multi-objective optimization," *IEEE Transactions on Signal Processing*, vol. 61, pp. 4416–4427, Sept 2013.

[119] M. T. Harandi, C. Sanderson, R. Hartley, and B. C. Lovell, "Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach," in *Proceedings of the 12th European Conference on Computer Vision - Volume Part II*, ECCV'12, (Berlin, Heidelberg), pp. 216–229, Springer-Verlag, 2012.

[120] E. C. Corts and C. Scott, "Sparse approximation of a kernel mean," *IEEE Transactions on Signal Processing*, vol. 65, pp. 1310–1323, March 2017.

[121] S. Wilson and C. K. Mohan, "An information bottleneck approach to optimize the dictionary of visual data," *IEEE Transactions on Multimedia*, vol. 20, pp. 96–106, January 2018.

[122] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Transactions on Signal Processing*, vol. 58, pp. 1553–1564, March 2010.

[123] J. Sulam, B. Ophir, M. Zibulevsky, and M. Elad, "Trainlets: Dictionary learning in high dimensions," *IEEE Transactions on Signal Processing*, vol. 64, pp. 3180–3193, June 2016.

[124] C. K. I. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems 13* (T. K. Leen, T. G. Dietterich, and V. Tresp, eds.), pp. 682–688, MIT Press, 2001.

[125] K. Zhang, I. W. Tsang, and J. T. Kwok, "Improved nystrÖm low-rank approximation and error analysis," in *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, (New York, NY, USA), pp. 1232–1239, ACM, 2008.

[126] D. Feldman, M. Feigin, and N. Sochen, "Learning big (image) data via coresets for dictionaries," *J. Math. Imaging Vis.*, vol. 46, pp. 276–291, July 2013.

[127] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit," *CS Technion*, vol. 40, pp. 1–15, April 2008.

[128] S. Mukherjee and C. S. Seelamantula, "A divide-and-conquer dictionary learning algorithm and its performance analysis," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4712–4716, March 2016.

[129] D. Barchiesi and M. D. Plumbley, "Learning incoherent dictionaries for sparse approximation using iterative projections and rotations," *IEEE Transactions on Signal Processing*, vol. 61, pp. 2055–2065, April 2013.

[130] M. Nejati, S. Samavi, S. M. R. Soroushmehr, and K. Najaran, "Coherence regularized dictionary learning," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4717–4721, March 2016.

[131] B. Mailh, D. Barchiesi, and M. D. Plumbley, "Ink-svd: Learning incoherent dictionaries for sparse representations," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3573–3576, March 2012.

[132] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[133] K. K. Reddy and M. Shah, "Recognizing 50 human action categories of web videos," *Mach. Vision Appl.*, vol. 24, pp. 971–981, July 2013.

[134] S. Wilson, M. Srinivas, and C. Mohan, "Dictionary based action video classification with action bank," in *Digital Signal Processing (DSP), 2014 19th International Conference on*, pp. 597–600, Aug 2014.

[135] B. Solmaz, S. M. Assari, and M. Shah, "Classifying web videos using a global video descriptor," *Mach. Vision Appl.*, vol. 24, pp. 1473–1485, Oct. 2013.

[136] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf, "Motion interchange patterns for action recognition in unconstrained videos," in *Proceedings of the 12th European Conference on Computer Vision - Volume Part VI*, ECCV'12, (Berlin, Heidelberg), pp. 256–269, Springer-Verlag, 2012.

List of Publications

BOOK CHAPTER

1. Shyju Wilson, C. Krishna Mohan and K. Srirama Murthy,"Event Based Sports Videos Classification using HMM Framework," ***Computer Vision in Sports (Springer International Publishing)***, pp. 229-244, 2014.

JOURNALS

1. Shyju Wilson and C. Krishna Mohan, "An information bottleneck approach to optimize the dictionary of visual data", ***IEEE Transactions on Multimedia,*** vol. 20, no. 1, pp. 96-106, January 2018.

2. Shyju Wilson and C. Krishna Mohan, "Coherent and non-coherent dictionary for action recognition", ***IEEE Signal Processing Letters,*** vol. 24, no. 5, pp. 698-702, May 2017.

3. Shyju Wilson and C. Krishna Mohan, "Information loss based sampling to linearize kernel dictionary learning", Communicated to ***IEEE Transactions on Signal Processing.*** (Under Review)

CONFERENCES

1. Shyju Wilson, M. Srinivas and C. Krishna Mohan, "Dictionary based action video classification with action-bank", in *Digital Signal Processing **(DSP 2014)**, International Conference on* , pp. 597-600, Aug 2014.

CURRICULUM VITAE

1. **Name:** Shyju Wilson
2. **Date of Birth:** 6^{th} May 1984
3. **Permanent Address:**

   Shyju Nivas
   Sasthamcotta
   Kollam Dist.
   Kerala, India - 690521.

4. **Educational Qualifications:**

   - Dec 2017: Doctor of Philosophy in Computer Science and Engineering,
     IIT Hyderabad, Telangana, India.
   - Apr 2009: Master of Technology in Computer Science and Engineering,
     NIT Rourkela, Orissa, India.
   - Sep 2006: Bachelor of Technology in Computer Science and Engineering,
     Institution of Engineers (India), Kolkata, India.

DOCTORAL COMMITTEE

1. **Chairperson:** Dr. M. V. Panduranga Rao

2. **Guide:** Dr. C. Krishna Mohan

3. **Members:**

   - Dr. M. V. Pandurangarao (Dept. of CSE)
   - Dr. Sobhan Babu (Dept. of CSE)
   - Dr. Sathya Peri (Dept. of CSE)
   - Dr. Sri Rama Murty (Dept. of EE)