K-SVD BASED DICTIONARY LEARNING ALGORITHMS FOR ACCELEROMETER SIGNAL PROCESSING

Thesis Submitted in the Partial Fulfilment

of the Requirements for the Degree of

MASTER OF ELECTRICAL ENGINEERING

by

ASHWIN RAI

(Reg. No. **160167 of 2021-22)**

(Exam Roll No. M4ELE23003)

Under the guidance of

Dr. AMITAVA CHATTERJEE

Electrical Engineering Department

Jadavpur University, Kolkata, India

2021-2023

K-SVD BASED DICTIONARY LEARNING ALGORITHMS FOR ACCELEROMETER SIGNAL PROCESSING

Thesis Submitted in the Partial Fulfilment of the Requirements for the Degree of

MASTER OF ELECTRICAL ENGINEERING

by

ASHWIN RAI

Class Roll No.: **002110802011**

Registration No.: **160167 OF 2021-22**

JADAVPUR UNIVERSITY

Under the guidance of

Prof. (Dr.) AMITAVA CHATTERJEE

Electrical Engineering Department

Faculty of Engineering and Technology

JADAVPUR UNIVERSITY

KOLKATA- 700032, W.B. INDIA

2021-2023

JADAVPUR UNIVERSITY

FACULTY OF ENGINEERING AND TECHNOLOGY

ELECTRICAL ENGINEERING DEPARTMENT

KOLKATA- 700032, INDIA

CERTIFICATE OF RECOMMENDATION

This is to certify that, Mr. ASHWIN RAI (Registration No. 160167 of 2021-22), has completed and submitted his thesis entitled, **"K-SVD BASED DICTIONARY LEARNING ALGORITHMS FOR ACCELEROMETER SIGNAL PROCESSING"**, in partial fulfilment of the requirements for the degree of "Master of Electrical Engineering" of Jadavpur University. The thesis work has been carried out by him under my guidance and supervision. The project, in my opinion, is worthy of its acceptance.

SUPERVISOR

Prof. (Dr.) Amitava Chatterjee

Professor
Electrical Engineering Department,

Jadavpur University

COUNTERSIGNED

Prof. (Dr.) Biswanath Roy

Head, Dept. of Electrical Engineering,

Faculty of Engineering and Technology,

Jadavpur University

Prof. (Dr.) Ardhendu Ghoshal

Dean,

Faculty of Engineering and Technology,

Jadavpur University

JADAVPUR UNIVERSITY

KOLKATA-700032, INDIA

FACULTY OF ENGINEERING AND TECHNOLOGY

ELECTRICAL ENGINEERING DEPARTMENT

CERTIFICATE OF APPROVAL*

The foregoing thesis is hereby approved as a creditable study of ***Master of Electrical Engineering*** *under Electrical Engineering department and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion therein but approve this thesis only for the purpose for which it is submitted.*

Final Examination for Evaluation of the Thesis

1. ______________________________

2. ______________________________

(Signature of Examiners)

*Only in case the thesis is approved

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that the thesis entitled "**K-SVD Based Dictionary Learning Algorithms for Accelerometer Signal Processing**" contains literature survey and original research work as part of the course of Master of Engineering under Electrical Engineering department. All the information in this document have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name	:	**ASHWIN RAI**
Class Roll No.	:	**002110802011**
Registration No.	:	**160167 of 2021-22**
Thesis Name	:	**K-SVD BASED DICTIONARY LEARNING ALGORITHM FOR ACCELEROMETER SIGNAL PROCESSING**
Signature with date	:	

ACKNOWLEDGEMENTS

With a great pleasure, I express my deep sense of gratitude to my supervisor, **Prof. Amitava Chatterjee,** Department of Electrical Engineering of Jadavpur University for giving me the invaluable opportunity to work in this exciting domain of research. I am obliged and grateful to him for his guidance and suggestions throughout the project work. His appreciation and encouragement throughout the project, had always been a confidence booster for me. I feel extremely lucky and shall always be indebted to him, for having him as my thesis supervisor.

I would also wish to express my sincere gratitude to **Prof. Sugata Munshi, Prof. Debangshu Dey, Prof. Biswajit Bhattacharyya, Prof. Mita Dutta, Prof. Gautam Sarkar,** and **Prof. Palash Kr. Kundu** Department of Electrical Engineering, Jadavpur University, for their encouragement, advice and motivation during the coursework.

I am also thankful to **Prof. Biswanath Roy**, Head, Department of Electrical Engineering, Jadavpur University, for providing the necessary facilities to carry out this research work.

I am taking the opportunity to express my humble indebtedness to **Mrs. Pubali De**, Research Scholar, Jadavpur University, for her invaluable inputs during this work. I am also thankful to rest of the research scholars of Measurement laboratory, e.g., **Mr. Soibal Ghosh**, **Mrs. Antara Ghosh**, **Mrs. Sayanti Chaudhuri**, **Mr. Anadi Biswas**, and others for their support throughout the tenure of my research work.

I would like to thank my dear friends **Mr. Aninda Sundar Mondal** and **Mr. Rohan Das**, PG scholar, E.E. Department, from whom I received immense support, inexplicable encouragements and assistance throughout the time. I would like to convey my benevolent thankfulness to the rest of the PG scholars of E.E. Department for their moral support during this course work. Finally, I express my foremost gratefulness towards my parents for their constant support and motivation, without whom I would not have come so far.

Above all, I thank the almighty, which I believe to be none other than the nature itself, for keeping me in a good and sound condition, to be able to complete this work.

Thanking all,

Ashwin Rai

ABSTRACT

K-SVD is a very powerful dictionary learning algorithm which is used in signal processing and machine learning. The learned dictionary can then be used for classification problem. Human Activity Recognition is one of the very important classification-problem. HAR has various application such as Healthcare and Assistive Technologies, Ambient Assisted Living, Security and Surveillance, Fitness and Sports Tracking etc.

Human Activity Recognition is a process wherein the activity performed by human is analyzed and classified into its respective classes. The dictionary learning process for the HAR problem is done by algorithms such as K-SVD, MOD (Method of Optimal Direction, Maximum Likelihood Method etc. In this thesis, we have extensively discussed about the K-SVD based dictionary learning algorithms.

K-SVD based dictionary learning algorithms are variants of the original K-SVD algorithm such as Discriminative K-SVD, Approximate K-SVD, Block K-SVD, Parallel K-SVD, Structured K-SVD etc. Although K-SVD is a very powerful dictionary learning algorithm on its own, with ever increasing dictionary size for complex human activity recognition, the computational burden increases drastically and will consume more time to converge to a solution. Therefore, the algorithm must not only be powerful but also should have less computational burden. The variants of the original K-SVD algorithm aims to converge to a solution faster and improve efficiency as well.

One of the variants is Discriminative K-SVD. As the name suggest, this algorithm helps in learning a dictionary with not only representational power but adds a discriminative power to the dictionary as well. The dictionary learned with a discriminative power is more suitable for the application in classification problem. This variant tends to increase the efficiency of the classification problem.

Approximate K-SVD is another variant of the K-SVD based dictionary learning. This variant tends to decrease the computational burden by approximating the solution without calculating the actual solution. This helps the algorithm to be executed much faster. However, the accuracy of the dictionary learning process may get slightly lowered.

The sparse coding is also an integral part of the dictionary learning process. There are various sparse coding algorithms such as OMP (Orthogonal Matching Pursuit), Basic Pursuit, Focal Underdetermined System Solver (FOCUSS) etc. In this thesis, we have discussed five different sparse coding algorithms. They are OMP, Batch OMP, POMP (Projection Based Orthogonal

Matching Pursuit), OLS (Orthogonal Least Square) and LAOLS (Look Ahead Orthogonal Least Square).

The K-SVD algorithm and its variant (D-KSVD and A-KSVD) is implemented with five different sparse coding algorithms to solve a bi-class classification problem of human activity recognition on 3 different datasets. All these has been covered in this thesis. The future scope on this topic has also been discussed in the conclusion.

LIST OF CONTENTS

LIST OF ABBREVIATIONS

ABBREVIATION	EXPLICATION
1. MOD	Method of Optimal Direction
2. AAL	Ambient Assisted Living
3. DL	Dictionary Learning
4. FOCUSS	Focal Underdetermined System Solver
5. CoSaMP	Compressive Sampling Matching Pursuit
6. OMP	Orthogonal Matching Pursuit
7. POMP	Projection based Orthogonal Matching Pursuit
8. OLS	Orthogonal Least Square
9. LAOLS	Look Ahead Orthogonal Least Square
10. EOG	Electrooculography
11. SVD	Singular Value Decomposition
12. EM	Expectation Maximization
13. MAP	Maximum A Posteriori
14. VQ	Vector Quantization
15. PCA	Principle Component Analysis
16. HMP	Human Motion Primitives
17. LASSO	Least Absolute Shrinkage and Selection Operator
18. IST	Iterative Soft Thresholding
19. IMU	Inertial Measurement Unit

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

- BACKGROUND
- MOTIVATION OF THE THESIS
- CONTRIBUTIONS IN THE THESIS
- STRUCTURE OF THE THESIS

1.1. BACKGROUND

Dictionary is set of atoms which is used to represent a data in sparse and concise manner. The data is represented as a linear combination of few atoms of the dictionary. Dictionary itself is a matrix. Each column vector of the Dictionary matrix is known as atom. The dictionary can be designed in two ways. First, selecting one from prespecified set of linear transforms and second, adapting a dictionary to a set of training signals [1]. The simplicity of selecting a dictionary from a predetermined set of linear transforms makes it appealing. In many instances, it even results in straightforward and quick methods for the sparse representation assessment. For adapting a dictionary to a set of training signals we use novel algorithm approach. Using a novel algorithm, dictionaries are adjusted to a set of training signals in order to produce sparse signals. We look for the dictionary that, given a collection of training signals, leads to the best representation for each member of the set while strictly adhering to sparsity restrictions [1].

Dictionary learning is a matrix factorization technique in which the signal $'\mathbf{y}'$ can be represented as linear multiplication of Dictionary atoms and sparse matrix $'\mathbf{x}'$. Mathematically, it can be represented as follows [1],

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

Various algorithms have been employed to solve the problem stated by (1.1). Some of them are K-SVD [1][25], MOD (Method of Optimal Direction) [1] [38],[39],[40], Maximum Likelihood Method [1] [41],[42],[43],[44] etc.

1.2. MOTIVATION OF THE THESIS

Dictionary Learning finds its application in various fields such as image processing, audio processing, signal processing. One application that is very helpful in our daily life is human movement detection. Human movement detection is used in Ambient Assisted Living (AAL) [45],[46],[47], security surveillance [49] etc.

The acceleration data produced during human activities can be used to classify the type of activities performed by the human by comparing them with the training signals. Training signals are set of data that was once acquired and stored. Now these data serve as the initial dictionary which is learned using dictionary learning algorithms and made fit for classification problem. Such classification is very helpful in developing various technologies for AAL, security surveillance etc. The accelerometer sensor-based data provide a rich variety of signal or data information which can be advantageously utilized to solve many recognition problems. This served as a motivation to explore within this thesis how different datasets acquired using accelerometer sensor can be effectively classified using different variants of K-SVD based dictionary learning (DL) algorithms and how the performances of such DL algorithms can vary with different choices of algorithms to solve the sparse coding stage.

1.3. CONTRIBUTIONS IN THE THESIS

There are various algorithms for Dictionary Learning. K-SVD based dictionary algorithms is one of the algorithms used for dictionary learning. There have been many citations of K-SVD based dictionary algorithms (K-SVD [1],[20],[21],[22],[23],[24], Discriminative-KSVD [29],[30],[32] and Approximate-KSVD [3]) which have been used for dictionary learning.

The dictionary learning algorithm has 2 stages. First, there is Sparse Coding process. There are different sparse coding algorithms being used, such as Basic Pursuit [1],[50], FOCUSS (Focal Underdetermined System Solver) [1],[52],[53],[54], CoSaMP (Compressive Sampling Matching Pursuit) [55],[56], etc. Second, dictionary update stage. K-SVD is probably the most traditional and popular algorithm proposed so far for dictionary learning (DL) [1],[20],[21],[22],[23],[24]. In this thesis, several variations of K-SVD algorithm have been introduced and implemented and studied in detail, for a variety of problems at hand. For the sparse coding algorithms, this thesis investigates the effectiveness of five different algorithms viz., OMP (Orthogonal Matching Pursuit), Batch OMP, POMP (Projection Based OMP), OLS (Orthogonal Least Square) and LAOLS (Look Ahead Orthogonal Least Square), in combination with the K-SVD, D-KSVD and A-KSVD. These three DL algorithms with K-SVD variants and introducing five different algorithms for sparse coding stage in each such DL algorithm have been finally implemented for three different accelerometer data-based datasets and their performance evaluations are carried out in a systematic manner. The three different datasets are as follows

i. Human Behavior detection based on a Motion Primitives Detection Dataset
ii. Human Activity detection based on a Single Chest-Mounted Accelerometer Dataset
iii. Classification of Road and Types using a dataset created by wearing EOG Smart Glasses and an Inertial Measurement Unit, comprising gyroscope and accelerometer

There are many other very interesting aspects which could have been further studied e.g., the number of K-SVD variants and within each K-SVD variant the number of sparse coding algorithms studied could have been further enhanced. Similarly, the number of datasets under investigation also could have been further expanded. However, considering the time-bounded constraints associated with a Master's Thesis, those aspects have been left for future scope of work.

1.4. STRUCTURE OF THE THESIS

In this thesis, the literature of the K-SVD based dictionary learning algorithms have been discussed and the same have been applied on three different datasets. The dictionary learning algorithms discussed in this thesis are K-SVD, Discriminative K-SVD (D-KSVD), Approximate K-SVD (A-KSVD). One important part of the dictionary learning process is sparse coding. There are various sparse coding algorithms present. However, in this thesis, five different types of sparse coding algorithm have been covered. They are OMP (Orthogonal Matching Pursuit), Batch OMP, POMP (Projection based OMP), OLS (Orthogonal Least Square) and LAOLS (Look Ahead Orthogonal Matching Pursuit).

Chapter 2 deals with the literature of K-SVD Algorithm and all five sparse coding algorithms mentioned above. D-KSVD is discussed in chapter 3 and A-KSVD is discussed in Chapter 4. In Chapter 5, there is application of K-SVD, D-KSVD, A-KSVD algorithms in combination with all five sparse coding algorithms on Human Motion Primitives Detection Dataset. The results have been compiled and elaborated in this chapter. Similarly, in chapter 6, we have application of K-SVD, D-KSVD and A-KSVD in combination with all five sparse coding algorithms on Activity Recognition from a Single Chest-Mounted Accelerometer Dataset. The results have been compiled and elaborated in this chapter. Chapter 7 has application of K-SVD, D-KSVD and A-KSVD in combination with all five sparse coding on Dataset for Classification of Road and Types using EOG Smart Glasses. The results have been compiled and elaborated in this chapter. Finally, we conclude the thesis by writing a conclusion in chapter 8.

CHAPTER 2

K-SVD ALGORITHM FOR DICTIONARY LEARNING

- INTRODUCTION
- MOTIVATION
- K-SVD ALGORITHM
  - K-means ALGORITHM
  - K-SVD ALGORITHM
  - SPARSE CODING
  - DICTIONARY UPDATE
- SUMMARY

2.1. INTRODUCTION

The K-SVD dictionary learning algorithm uses a singular value decomposition (SVD) method to build a dictionary for sparse representations [1] [20]. It uses an extension of SVD to find an over-complete dictionary from a collection of training data, iteratively switching between sparse-coding the input data based on the current dictionary and changing the dictionary's atoms to better fit the data is how the K-SVD generalization of the K-means clustering algorithm operates. It shares structural similarities with the algorithm for expectation maximization (EM) [57].

A statistical algorithm known as the expectation-maximization (EM) algorithm is an iterative technique for determining (local) maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models where the model depends on latent variables that are not observed [57]. The EM iteration alternates between performing an expectation (E) step, which generates a function for the expectation of the log-likelihood evaluated using the most recent estimate for the parameters, and a maximization (M) step, which computes the parameters maximizing the expected log-likelihood discovered on the E step. The distribution of the latent variables is subsequently ascertained in the following E phase using these parameter-estimates.

Applications like image processing, audio processing, machine learning and signal processing all frequently use the K-SVD algorithm [1],[20],[21],[22],[23],[24].

The fundamental idea behind K-SVD is to linearly combine a few atoms (columns) from the learnt dictionary to represent each signal in the training data. By resolving an optimization problem that aims to reduce the error between the original data and the reconstructed data using the dictionary, the dictionary is learned.

The optimization problem under consideration can be given as [1]:

$$\min_{\mathbf{D},\mathbf{X}} \{\| \mathbf{Y} - \mathbf{D}\mathbf{X} \|_F^2\} \; \textit{subject to} \; \forall_i, \;\; \| \mathbf{x}_i \|_0 \leq T_0 \tag{2.1}$$

The sparse representation of the signals in the training data is updated alternately with updating the dictionary as the K-SVD method advances iteratively. The algorithm eventually produces a dictionary that is designed to sparsely and effectively represent the training data.

2.2. MOTIVATION

So far, a very wide range of applications have seen successful applications of K-SVD based dictionary learning algorithms. For instance, as mentioned before, it finds its application in image processing, audio processing, machine learning and signal processing [1],[20],[21],[22],[23],[24]. It is possible to learn a concise representation of data using K-SVD. For instance, using K-SVD, we may develop a dictionary of basic pictures that can be applied to represent the photos in a much more condensed manner when we have a huge dataset of images.

By learning a dictionary of basis vectors that capture the underlying structure of the signal, K-SVD can be used to denoise signals. By determining the sparsest representation of the signal in terms of the dictionary, we may utilize the learnt dictionary to denoise a noisy signal.

For classification problems, K-SVD can be used to learn a discriminative dictionary. We can enhance the effectiveness of classifiers by lowering the dimensionality of the feature space by learning a dictionary of basis vectors that capture the relevant structure of the data.

K-SVD is a strong tool for learning basis vector dictionaries and can be applied to a variety of signal processing and machine learning problems.

2.3. K-SVD ALGORITHM

Any pursuit algorithm can be used in conjunction with this flexible algorithm. It is straightforward and intended to be a true generalization of the K-means. As a result, it learns a dictionary for the gain-shape VQ (Vector Quantization) when compelled to operate with one atom per signal. This atom must have a unit coefficient in order for the algorithm to accurately duplicate the K-means method. Due to efficient sparse coding and an accelerated dictionary updating mechanism that resembles Gauss-Seidel, the K-SVD has a high level of efficiency. The algorithm's phases are consistent with one another and both aim to minimize a distinct overall objective function [1].

Since the K-SVD algorithm is the generalization of the K-means algorithm, let us first discuss the K-means algorithm.

2.3.1. K-means ALGORITHM

'K' in the K-means clustering algorithm denotes the number of clusters that the algorithm should attempt to find in the provided dataset. The K-means algorithm is an unsupervised learning technique used to cluster or divide related data points in a dataset into a collection of clusters called 'K'.

The initial cluster centers for the K-means method are chosen at random from the dataset as 'K' points (also known as centroids). Once each data point has been assigned to its closest centroid, the algorithm calculates new centroids by taking the mean of all the data points in that cluster. Up to a certain number of iterations or until the centroids stop moving, this process is done repeatedly.

Typically, the ideal number of clusters 'K' is chosen by the user or by an analysis of the data. Smaller values of 'K' produce bigger, more generalized clusters, whereas greater values of 'K' produce smaller, more fine-grained clusters. The initial centroids selected for the K-means algorithm are crucial to understand since several random initializations are frequently employed to increase the system's robustness.

Let us define a codebook matrix by C, where $\mathbf{C} = [\mathbf{c_1}, \mathbf{c_2} ... \mathbf{c_K}]$. The columns in the codebook are codewords. There are K number of codewords in a codebook to represent a wide numbers of vector family using nearest neighbor assignment. The vector family is denoted by [1]:

$$\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N} \ (N \gg K) \tag{2.2}$$

When the $\mathbf{C}$ is known, each signal from the training dataset, i.e., $\mathbf{Y}$, is represented as the closest codeword that it resembles. This representation is done under l_2-norm distance. Hence, we can write [1]:

$$\mathbf{y}_i = \mathbf{C}\mathbf{x}_i \tag{2.3}$$

Here, $\mathbf{x}_i = \mathbf{e}_j$, is a vector from trivial basis with non-zero entry at j^{th} position only. The j^{th} index is selected such that [1]:

$$\forall_{k \neq j} \ \| \mathbf{y}_i - \mathbf{C}\mathbf{e}_j \|_2^2 \ \leq \ \| \mathbf{y}_i - \mathbf{C}\mathbf{e}_k \|_2^2 \tag{2.4}$$

Given that just one atom is permitted to participate in the building of $\mathbf{y}$ and that the coefficient must be 1, this is regarded as an extreme case of sparse coding. The Mean Square Error (MSE) for each signal of the training datasets is given by [1],[26]:

$$\mathbf{e}_i^2 = \| \mathbf{y}_i - \mathbf{C}\mathbf{x}_i \|_2^2 \tag{2.5}$$

And the overall MSE is given by [1]:

$$\mathbf{E} = \sum_{i=1}^{K} \mathbf{e}_i^2 = \| \mathbf{Y} - \mathbf{C}\mathbf{X} \|_F^2 \tag{2.6}$$

The main objective here is to derive a codebook $\mathbf{C}$ which minimizes the mean square error E, such that the structure of the $\mathbf{X}$ is limited (i.e., $\mathbf{X}$ is sparse). This can be given as:

$$\min_{\mathbf{C},\mathbf{X}} \{\Omega \| \mathbf{Y} - \mathbf{C}\mathbf{X} \|_F^2\} \text{ subject to } \forall_i, \mathbf{x}_i = \mathbf{e}_k \text{ for some } k \tag{2.7}$$

The K-means algorithms used to solve the problem given below to find the best codebook to represent the signals from training dataset given by (2.2) can be seen as: [1]

Initialization: Set the codebook matrix $\mathbf{C}^{(0)} \in \Re^{n \times K}$ and $J = 1$. Repeat until convergence using stop rule.
Stage 1: Sparse coding stage
Partition the training samples $\mathbf{Y}$ into K sets
$(\mathbf{R}_1^{(J-1)}, \mathbf{R}_2^{(J-1)}, \ldots, \mathbf{R}_K^{(J-1)})$
Each holding sample indices most similar to the column $\mathbf{c}_k^{(J-1)}$,
$\mathbf{R}_K^{(J-1)} = \{ i \mid \forall_{l=k}, \| \mathbf{y}_i - \mathbf{c}_k^{(J-1)} \|_2 \ < \ \| \mathbf{y}_i - \mathbf{c}_l^{(J-1)} \|_2 \}$
Stage 2: Codebook update stage

For each column k in $\mathbf{C}^{(J-1)}$, update it by

$$\mathbf{c}_k^{(J)} = \frac{1}{|\mathbf{R}_k|} \sum_{i \in \mathbf{R}_k^{(J-1)}} \mathbf{y}_i$$

Set $J = J + 1$

Fig. 2.1. K-means Algorithm [1].

The best codebook for VQ is created iteratively using the k-means technique. Each iteration consists of two stages: the sparse coding stage, which essentially evaluates $\mathbf{X}$, and the codebook updating stage.

In the sparse coding stage, we consider a known codebook matrix $\mathbf{C}$ and derive a matrix $\mathbf{X}$ such that it minimizes the value of (2.7). In the Codebook update stage, we consider the previously calculate $\mathbf{X}$ as known matrix and find an updated codebook matrix $\mathbf{C}$, such that (2.7) is minimized. Each repetition ensures that the MSE either decreases or stays the same. Additionally, under the assumptions, the minimization step is the best one at each such stage. Convergence to at least a local minimum solution is assured since the MSE is constrained from below by zero and the algorithm ensures a monotonic decrease of the MSE

2.3.2. K-SVD ALGORITHM

In the K-SVD, a signal can be represented by a linear combination of codewords. The coefficient matrix vector may therefore have more than one non-zero entry. The codebook known as Dictionary and represented by the letter $\mathbf{D}$ in the K-SVD method.

The minimization problem to be solved for K-SVD is hence given by

$$\min_{\mathbf{D},\mathbf{X}} \{\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2\} \text{ subject to } \forall_i, \|\mathbf{x}_i\|_0 < T_0 \tag{2.8}$$

The same solution can be obtained by solving the equation

$$\min_{\mathbf{D},\mathbf{X}} \sum_i \|\mathbf{x}_i\|_0 \text{ subject to } \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \le \varepsilon \tag{2.9}$$

Our initial goal is to fix **D** and determine the best coefficient matrix $\mathbf{X}$ that can be located. Since it is impossible to find the truly ideal $\mathbf{X}$, we use pursuit algorithm. For the computation of the coefficients, any such technique may be employed as long as it can produce a solution with a fixed and predetermined number of nonzero entries T_0.

After completing the work of sparse coding, a subsequent stage is carried out to look for a better dictionary. In this stage, with the exception of column $\mathbf{d}_k$, all columns in $\mathbf{D}$ are fixed. And then the new coefficients are found which will update the column $\mathbf{d}_k$ such that the MSE is reduced. Each column of the dictionary is updated separately. A simple solution based on the singular value decomposition (SVD) exists for the issue of updating only one column at a time. Furthermore, by using more relevant coefficients for subsequent column updates, allowing a change in the coefficient values while updating the dictionary columns speeds up convergence. Overall, the outcome is remarkably similar to the transition from gradient descent to Gauss-Seidel methods in optimization.

The K-SVD algorithm to find the best dictionary to represent the signals of training dataset by solving the function given by (2.8) is given below [1]:

Step 1: Initialization

Set the dictionary matrix $\mathbf{D}^{(0)} \in \Re^{n \times K}$ with l_2 normalized columns. Repeat Step 2-4 until convergence or stopping criteria is attained.

Step 2: Sparse coding

calculate the representation vector x for each sample y using any pursuit technique, by approximating the solution of

$i = 1, 2, ..., \text{ N}, \ \min_{\mathbf{x}_i}\{\| \mathbf{y}_i - \mathbf{D}\mathbf{x}_i \|_2^2\}$ subject to $\| \mathbf{x}_i \|_0 \leq T_0$

Step 3: Codebook update stage

For each column k, $k = 1, 2, ..., K$ in $\mathbf{D}^{(J-1)}$ update it by

- Define the group of examples that use this atom, $\omega_k = \{i \mid 1 \leq i \leq N, \mathbf{x}_T^k(i) \neq 0$
- Compute the overall representation matrix, $\mathbf{E}_k$, by $\mathbf{E}_k = \mathbf{Y} - \sum_{j \neq k} \mathbf{d}_j \mathbf{x}_T^j$
- Restrict $\mathbf{E}_k$ by choosing only the columns corresponding to $\boldsymbol{\omega}_k$ and obtain $\mathbf{E}_k^R$

- Apply SVD decomposition $\mathbf{E}_k^R = \mathbf{U\Delta V}^T$. Choose the updated dictionary column $\mathbf{d}_k$ to be the first column of $\mathbf{U}$. Update the coefficient vector $\mathbf{x}_k^R$ to the first column of $\mathbf{V}$ multiplied by $\mathbf{\Delta(1,1)}$.

Step 4: Set $J = J + 1$

Fig 2.2. K-SVD Algorithm [1]

In the sparse coding stage, the dictionary $\mathbf{'D'}$ is assumed fixed. The above mentioned (2.8) is used for calculating the spare representation matrix $\mathbf{'X'}$. The penalty term is given by the expression

$$\| \mathbf{Y} - \mathbf{DX} \|_F^2 = \sum_{i=1}^{N} \| \mathbf{y}_i - \mathbf{Dx}_i \|_2^2 \tag{2.10}$$

As a result, the issue raised in (2.8) can be divided into N separate issues of the kind:

$$\min_{\mathbf{x}_i} \{ \| \mathbf{y}_i - \mathbf{Dx}_i \|_2^2 \} \text{ subject to } \| \mathbf{x}_i \| \leq T_0, \quad \text{for } i = 1, 2, \ldots, N \tag{2.11}$$

This problem can be solved using various pursuit algorithms which shall be discussed in the next section.

2.3.3. SPARSE CODING

In signal processing and machine learning, sparse coding is a method for representing signals or data using a limited number of basis functions or features. Finding a set of basis functions that can efficiently represent the data with the fewest non-zero coefficients is the aim of sparse coding. A signal or data point is represented in sparse coding as a linear combination of a number of basis functions, with the basis functions' coefficients being sparse or largely zero. Using methods like Principle Component Analysis (PCA) or dictionary learning, the basis functions are inferred from the data. Speech recognition, natural language processing, and picture and video processing are just a few of the many uses for sparse coding. It is especially helpful in applications where it is desirable to minimize the dimensionality of the data for effective processing and storage, such as with picture or video data. A noisy signal is divided into its sparse form via sparse coding, and the noise is then eliminated by thresholding the small coefficients. The sparse representation serves as the input to machine learning models like sparse autoencoders and sparse coding-based classifiers, which also use sparse coding.

Few of the many available sparse coding techniques used in this thesis are OMP (Orthogonal Matching Pursuit) Algorithm, Batch OMP (Batch Orthogonal Matching Pursuit) Algorithm, POMP (Projected Orthogonal Matching Pursuit) Algorithm, OLS (Orthogonal Least Square) Algorithm and LAOLS (Look Ahead Orthogonal Least Square) Algorithm.

2.3.3.1 ORTHOGONAL MATCHING PURSUIT (OMP) ALGORITHM

By iteratively choosing the dictionary atoms (columns) that are most correlated with the residual signal, OMP seeks to determine the best sparse approximation of a signal. OMP begins with an initial residual signal and chooses the atom with the highest inner product with the current residual at each iteration. The chosen atom is added to the support group, and the least-squares solution is used to update the relevant coefficient. Until a stopping criterion, such as a maximum number of iterations or a specific threshold of the residual norm, is satisfied, this process is repeated.

One benefit of OMP is that it offers a quick and effective technique to approximate a signal in a sparse manner, especially when the dictionary is redundant or overcomplete. An overcomplete dictionary means it has more number of columns than rows. Additionally, under specific circumstances, such as when the dictionary is orthogonal or when the signal has a sufficient number of nonzero coefficients, OMP has a theoretical guarantee of locating the precise sparse solution.

Data compression, pattern recognition, and image and audio processing are just a few of the applications where OMP has been applied [16],[17],[18]. It is frequently combined with other algorithms to provide more precise and effective sparse representations of signals, such as dictionary learning methods like K-SVD.

The algorithm to solve the problem given by $\min_{\mathbf{x}} \|\mathbf{x}\|_0$ subject to $\mathbf{y} = \mathbf{D}\mathbf{x}$ is given by [2]:

Parameters: We are given the matrix $\mathbf{D}$, the vector $\mathbf{y}$, and the error threshold ε_0.

Initialization: Initialize $k = 0$, and set

- The initial solution $x^0 = 0$.
- The initial residual $\mathbf{r^0 = y - Dx^0 = y}$.
- The initial solution $Support\ \mathbf{S}^0 = Support\ \{\mathbf{x^0}\} = \phi$.

Main Iteration: Increment k by 1 and perform the following steps:

- Sweep: Compute the errors $\varepsilon(j) = \min_{\mathbf{z}_j} \|\mathbf{d}_j \mathbf{z}_j - \mathbf{r}^{k-1}\|_2^2$ for all j using the optimal choice $\mathbf{z}_j^* = \mathbf{d}_j^T \mathbf{r}^{k-1} / \|\mathbf{d}_j\|_2^2$.
- Update support: Find minimizer, j_0 of $\varepsilon(j) : \forall j \notin S^{k-1}, \varepsilon(j_0) < \varepsilon(j)$, and update $S^k = S^{k-1} \cup \{j_0\}$.
- Update provisional Solution: Compute $\mathbf{x}^k$, the minimizer of $\|\mathbf{Dx - y}\|_2^2$ subject to $Support\ \{\mathbf{x}\} = \mathbf{S}^k$.
- Update residual: Compute $\mathbf{r}^k = \mathbf{y - Dx}^k$.
- Stopping rule: If $\|\mathbf{r}^k\|_2 < \varepsilon_0$, stop. Otherwise, apply another iteration.

Output: The proposed solution is $\mathbf{x}^k$ obtained after k iterations.

Fig 2.3. OMP Algorithm [2]

In the given problem, we are solving to determine a sparsely represented vector $'\mathbf{x}'$ using the OMP Algorithm. We are assuming that dictionary $'\mathbf{D}'$ and signal $\mathbf{y}$ is obtained from a source or chosen at random. By using this algorithm, we are to finalize a value of $'\mathbf{x}'$ such that the equation $\mathbf{Dx = y}$ is satisfied.

Initially set $k = 0$ and then make the increment by 1.

Initially, we take the value of $'\mathbf{x}'$ to be zero, i.e., before the iteration starts, $\mathbf{x}^0$ (elements of $\mathbf{x}$ under initial situation) is assigned as a zero vector. The residual is basically an error obtained by the equation, $\mathbf{r = y - Dx}$. This residual $'\mathbf{r}'$ is later on used to update the $'\mathbf{x}'$ and also to maintain a sparsity constraint. Under initial situation, the value of $'\mathbf{x}'$ is zero, therefore, the residual under initial situation will be $\mathbf{r}^0 = \mathbf{y}$, i.e., the signal itself.

In the next step we calculate the correlation between the dictionary $'\mathbf{D}'$ and the residual $'\mathbf{r}'$ and find the position of the element which has maximum contribution. In simpler words, finding the column of $'\mathbf{D}'$ which has largest correlation or projection with $'\mathbf{r}'$ (Before 1^{st} iteration $\mathbf{r}^0 = \mathbf{y}$ and after 1^{st} iteration the residual $\mathbf{r}$ will subsequently be updated). That will be the position where a nonzero element will exist in the sparse matrix $\mathbf{x}$.

We calculate the sparse matrix $\mathbf{X}$ by multiplying the pseudoinverse of the Dictionary Matrix $\mathbf{D}$ taking into consideration the atom (column) which has maximum contribution obtained from previous calculation and the signal itself. The residual is than updated, $\mathbf{r}^1 = \mathbf{y} - \mathbf{D}\mathbf{x}^1$. We also calculate the least square error and compare it with a threshold value. If the least square error is below a threshold value, i.e., closer to the zero, than the iteration will break and we will have the sparse matrix $\mathbf{x}$. However, if the least square is not below the threshold, then the iteration is continued to find a proper matrix element which shall give the least square error below the threshold.

This process will continue till the number of nonzero elements specified in the sparse matrix vector $'\mathbf{x}'$ is reached.

2.3.3.2. BATCH ORTHOGONAL MATCHING PURSUIT (BATCH OMP) ALGORITHM

A variation of the basic OMP technique for recovering sparse signals is batch OMP (Orthogonal Matching Pursuit). Batch OMP picks a group of atoms at once to speed up the procedure, as opposed to regular OMP, which chooses one atom (column) at a time to add to the support set [3].

With Batch OMP, the computation of the inner products between the dictionary and the residual signal is done in parallel. Batch OMP picks a group of atoms that are best connected with the residual signal as opposed to picking one at a time. The user sets the batch size and can adjust it to strike a balance between accuracy and speed.

Following the selection of a group of atoms, a least-squares solution is used to calculate the corresponding coefficients. After that, the procedure is repeated using the updated residual signal until the desired degree of sparsity is attained.

When the dictionary or the number of measurements is high, batch OMP may be quicker than the conventional OMP method. Additionally, it can lessen the standard OMP algorithm's bottleneck, which is the computational cost of computing inner products between the dictionary and the residual.

When handling large sets of data, the computational burden can be lowered in Batch OMP as compared to regular OMP. In the OMP, the atom selection process requires to calculate the residual $\mathbf{'r'}$ and multiply it with $\mathbf{D}^T$. In the Batch OMP, it is possible to calculate $\mathbf{D}^T\mathbf{r}$ as a whole rather than calculating residual $\mathbf{'r'}$ and multiplying it with $\mathbf{D}^T$. The calculation of $\mathbf{D}^T\mathbf{r}$ directly can be achieved as follows.

Let us denote $\boldsymbol{\alpha} = \mathbf{D}^T\mathbf{r}$, $\boldsymbol{\alpha}^0 = \mathbf{D}^T\mathbf{y}$ and $\mathbf{G} = \mathbf{D}^T\mathbf{D}$ [3]:

$$\begin{aligned}\boldsymbol{\alpha} &= \mathbf{D}^T(\mathbf{y} - \mathbf{D_I}(\mathbf{D_I})^{\dagger}\mathbf{y}) \\ &= \boldsymbol{\alpha}^0 - \mathbf{G_I}(\mathbf{D_I})^{\dagger}\mathbf{y} \\ &= \boldsymbol{\alpha}^0 - \mathbf{G_I}(\mathbf{D}_I^T\mathbf{D_I})^{-1}\mathbf{D}_I^T\mathbf{y} \\ &= \boldsymbol{\alpha}^0 - \mathbf{G_I}(\mathbf{G_{I,I}})^{\mathbf{-1}}\boldsymbol{\alpha}_I^0\end{aligned}$$

Since $\boldsymbol{\alpha}^0$ and $\mathbf{G}$ are already calculated, the calculation of $\boldsymbol{\alpha}$ i.e., $\mathbf{D}^T\mathbf{r}$ can be done directly without calculating residual $\mathbf{'r'}$. The new step requires only multiplication of the matrices which requires a lot less computational burden. The inversion of the matrix $(\mathbf{G_{I,I}})$ is done by using progressive Cholesky factorization.

It can be noted that the orthogonalization step $\mathbf{'(D_I)^{\dagger}y'}$ is not carried out due to its high associated computational burden. Rather, progressive Cholesky or QR update process is used which is much practical. Therefore, $(\mathbf{D}_I)^{\dagger}\mathbf{y} = (\mathbf{D}_I^T\mathbf{D}_I)^{-1}\mathbf{D}_I^T\mathbf{y}$.

The algorithm to obtain a sparse representation $\mathbf{x}$ is given below [3]:

Input: $\boldsymbol{\alpha}^0 = \mathbf{D}^T\mathbf{y}$, $\varepsilon^0 = \mathbf{y}^T\mathbf{y}$, $\mathbf{G} = \mathbf{D}^T\mathbf{D}$ and target error ε.

Output: Sparse representation $\mathbf{x}$ such that $\mathbf{y} \approx \mathbf{Dx}$.

Step 1: Initialize: $I := ()$, $\mathbf{L}^{-1} := [1]$, $\delta^0 := 0$, $n := 1$

Step 2: while $\varepsilon^{n-1} > \varepsilon$ do.

Step 3: $\hat{k} := Arg \max_k \{|\boldsymbol{\alpha}_k^{n-1}|\}$

Step 4: $\mathbf{g} = \mathbf{G}_{I^n, \hat{k}}$

if $n > 1$ then,

$\mathbf{w} :=$ solve for $\mathbf{w}$ $\{\mathbf{L}^{n-1}\mathbf{w} = \mathbf{g}_{I_n}\}$

Step 5: $\mathbf{L}^n := \begin{pmatrix} \mathbf{L}^{n-1} & 0 \\ \mathbf{w}^T & \sqrt{1-\mathbf{w}^T\mathbf{w}} \end{pmatrix}$

end if.

Step 6: $I^n := (I^{n-1}, \hat{k})$

Step 7: $\mathbf{c}^n :=$ solve for $\mathbf{c}$ $\{\mathbf{L}^n(\mathbf{L}^n)^T \mathbf{c} = \boldsymbol{\alpha}_{I^n}^0\}$

Step 8: $\boldsymbol{\beta}^n := \mathbf{G}_{I^n}\mathbf{c}^n$

Step 9: $\boldsymbol{\alpha}^n := \boldsymbol{\alpha}^0 - \boldsymbol{\beta}^n$

Step 10: $\delta^n = (\mathbf{c}^n)^T \boldsymbol{\beta}_{I^n}^n$

Step 11: $\varepsilon^n = \varepsilon^{n-1} - \delta^n + \delta^{n-1}$

Step 12: $n := n+1$

Step 13: end while

Step 14: $\mathbf{x} := 0$

Step 15: $\mathbf{x}_{I^n} := \mathbf{c}^n$

Fig 2.4. Batch OMP Algorithm [3].

Since the residual $'\mathbf{r}'$ is never calculated, the error based stopping criteria is very hard to employ. However, this challenge has been solved in [3] wherein, the error is updated by using the expression:

$$\varepsilon^n = \varepsilon^{n-1} - \delta^n - \delta^{n-1} \tag{2.12}$$

$$\text{Where, } \delta^n = (\mathbf{x}^n)^T \mathbf{G}\mathbf{x}^n,$$
$$\delta^{n-1} = (\mathbf{x}^{n-1})^T \mathbf{G}\mathbf{x}^{n-1}$$

The computation burden for calculation of δ is very low since the product of $\mathbf{G}\mathbf{x}^n = \mathbf{G}_I \mathbf{x}_I^n = \mathbf{G}_I (\mathbf{G}_{I,I})^{-1} \boldsymbol{\alpha}_I^0$ which has already been computed before during the calculation of $\boldsymbol{\alpha}$. The multiplication of $\mathbf{G}\mathbf{x}^n$ with $(\mathbf{x}^n)^T$ requires a very little amount of time.

2.3.3.3. PROJECTED ORTHOGONAL MATCHING PURSUIT (POMP) ALGORITHM

Projection Based OMP is also a modification of the Orthogonal Matching Pursuit (OMP) for computation of Sparse Representation of a signal. Projection step is added in this algorithm which enforces sparsity constraints on the computed coefficients. The fundamental goal of POMP is to find the best sparse approximation of a signal by repeatedly choosing the dictionary atoms (columns) that are most correlated with the residual signal, with the additional restriction that the coefficients be projected onto a sparsity constraint set at each iteration [4]. Even when the dictionary is not entirely orthogonal, this projection step makes sure that the coefficients stay sparse throughout the procedure.

A batch of atoms that is best correlated with the residual signal is chosen for each POMP iteration. The estimated coefficients are then projected onto a set of sparsity constraints using a thresholding function, and the related coefficients are updated using a least-squares solution. The thresholding function enforces the sparsity restriction by setting tiny coefficients to zero. Once a stopping requirement is reached, the process is done once more.

The POMP Algorithm for computation of sparse representation of a signal is given below [4]:

Input: $\mathbf{D}, \mathbf{y}, T_0, L$

Initialize: $r = \mathbf{y}, I = \phi$

For $k = 1 : s$

- Compute correlation with residual: $z = \mathbf{D}^{\mathbf{T}} r$
- Select indices τ of L largest $|z_j|$
- Compute potential solution: $\mathbf{x} = \mathbf{LS}(\mathbf{D}, \mathbf{y}, I \cup \tau)$

- Select largest element index: $i = \arg\max_{j \in \tau} |x_j|$
- Increase support: $I \leftarrow I \cup \{i\}$
- Compute new solution: $\mathbf{x} = \mathbf{LS}(\mathbf{D}, \mathbf{y}, I)$
- Update residual: $r = \mathbf{y} - \mathbf{D}_I \mathbf{x}_I$

Fig 2.5. POMP Algorithm [4].

The POMP algorithm uses a matching pursuit criterion to select a L number of atoms from Dictionary $'\mathbf{D}'$. Using these selected atoms, a least square solution is calculated for the support. The column of the solution with highest element is selected. A very large element directs to a very high likelihood for the position to be of true support. We may also say that POMP algorithm with $L = 1$ is similar to OMP Algorithm. [4],[28]

2.3.3.4. ORTHOGONAL LEAST SQUARE (OLS) ALGORITHM

OLS sparse coding is a type of sparse coding that pulls information from a dictionary to estimate the sparse coefficients of a signal in a linear combination of atoms. In OLS sparse coding, it is assumed that the dictionary is overcomplete, which means that there are more atoms than there are dimensions to the signal. The objective is to use a subset of the atoms from the dictionary to identify the signal's sparsest representation. Iteratively choosing the atom from the dictionary that is most connected with the residual signal and adding it to the active set is how the method operates [4]. The active set of atoms is then used to estimate the sparse coefficients using the OLS sparse coding method. Until the appropriate level of sparsity is reached, the procedure is repeated.

OLS sparse coding has the benefit of being computationally efficient and capable of handling big dictionaries. Additionally, it can handle noise and it can be applied to applications for compression and denoising [6],[7],[8],[10],[11]. The optimal sparse representation might not always be achieved, especially if the dictionary is not a good fit for the signal.

Applications for OLS include feature selection, pattern recognition, and images and audio processing [5],[9].

The OLS Algorithm for sparse coding is given below [4]:

Input: $\mathbf{D}, \mathbf{y}, T_0$

Initialize: $I = \phi$

for $k = 1 : T_0$ do

for $j \notin I$ do

- Build new support: $\tau = I \cup \{j\}$
- Try solution: $\mathbf{x} = LS(\mathbf{D}, \mathbf{y}, \tau)$
- Residual norm: $\rho_j = \| \mathbf{y} - \mathbf{D}_\tau \mathbf{x}_\tau \|^2$

Select new column: $i = \arg\min_j \rho_j$

Increase support: $I \leftarrow I \cup \{i\}$

Compute new solution: $\mathbf{x} = LS(\mathbf{D}, \mathbf{y}, I)$

Fig 2.6. OLS Algorithm [4].

In the OMP algorithm, first correlation is calculated between the atom of the dictionary and the residual. The index of the maximum correlation is added to the support. The residual for next iteration is then calculated using this support. The new column of the dictionary is selected in such a way that it reduces the residual however keeping the sparse solution fixed. However, in the OLS algorithm, the new column of the dictionary is selected in such a way that the new column along with the previous column shall give a reduced value of residual. In the process, the nonzero entries of sparse matrix can change.[4][27]

2.3.3.5. LOOK AHEAD ORTHOGONAL LEAST SQUARE (LAOLS) ALGORITHM

LAOLS (Look Ahead Orthogonal Least Square algorithm) sparse coding is a type of sparse coding that combines the benefits of OLS (Orthogonal Least Squares) with l_1 norm regularization to produce a sparse representation of a signal that is more reliable and accurate. The dictionary is assumed to be overcomplete i.e., the number of atoms of the dictionary is greater than the dimension of the signal, as is considered in other situations too.

The algorithm adds that atom from the active set which is most connected with the residual signal after iteratively choosing it from the dictionary. Then, using the OLS method on the active set of atoms and l_1 norm regularization on the full set of atoms, the sparse coefficients are computed

[4]. In order to prevent overfitting, the l_1 norm regularization promotes sparsity in the estimated coefficients.

The fact that LAOLS is computationally effective and capable of handling big dictionaries is one of its benefits. Additionally, it can handle noise and it can be applied for denoising and compression purposes. It has been demonstrated to be more accurate and reliable than OLS, particularly when the signal is heavily correlated or the dictionary is not a good match for the signal.

LAOLS finds its applications in various fields of feature selection, pattern identification, image and audio processing. [12],[13],[14],[15].

The LAOLS sparse coding algorithm for computing sparse representation is given below [4]:

Input: $\mathbf{D}, \mathbf{y}, T_0, L$

Initialize: $\mathbf{r} = \mathbf{y}, I = \phi$

for $k = 1:s$ do

- Compute correlations with residual: $\mathbf{z} = \mathbf{D}^T\mathbf{r}$
- Select indices τ of L largest $|\mathbf{z}_j|$
- for $j \in \tau$ do
    1. Run OMP starting from $I \cup \{j\}$, obtaining $\mathbf{x}$
    2. Compute residual norm $\rho_j = \|\mathbf{y} - \mathbf{Dx}\|^2$
- Select new column: $i = \arg\min_j \rho_j$
- Increase support: $I \leftarrow I \cup \{i\}$
- Compute new solution: $\mathbf{x} = LS(\mathbf{D}, \mathbf{y}, I)$

Fig 2.7. LAOLS Algorithm [4].

L number of atoms are selected from the Dictionary using matching pursuit criteria just as in POMP algorithm. The support is updated by new indices that we get from solving a least square problem for the selected L atoms of the dictionary and the residual. Then the OMP algorithm is run starting from this updated support to T_0, i.e., the sparsity constrained. The index that has just been chosen is the one that produces the least residual for the ultimate OMP solution. Hence, for selection of each index, a look ahead search is performed.

It can be noted that the number of selected atoms $'L'$ from the dictionary and sparsity constraint $'T_0'$ are user defined [4],[28].

2.3.4. DICTIONARY UPDATE

The next stage of the algorithm is the Dictionary update stage. First, we define a group of signals which use the atom in question as ω_k. This group indicates the position where there are nonzero entries in the $\mathbf{x}_T^k$ which will be required later for the calculation of the penalty term. ω_k is defined to later on maintain the sparsity of the $\mathbf{x}_T^k$ which otherwise would most likely be completely filled during updating process using Singular Value Decomposition (SVD).

In this stage both $\mathbf{D}$ (Dictionary) and $\mathbf{X}$ (Sparse Coefficient) are considered fixed. However, only one column of the dictionary, i.e., $\mathbf{d}_k^{th}$ column and the coefficients corresponding to it, i.e., $\mathbf{k}^{th}$ row of the $\mathbf{X}$ matrix is analyzed for dictionary update. From (2.8), the error term can be rewritten as [1],[25]:

$$\begin{aligned} \| \mathbf{Y} - \mathbf{DX} \|_F^2 &= \| \mathbf{Y} - \sum_{j=1}^{K} \mathbf{d}_j \mathbf{x}_T^j \|_F^2 \\ &= \left\| \left(\mathbf{Y} - \sum_{j \neq k} \mathbf{d}_j \mathbf{x}_T^j \right) - \mathbf{d}_k \mathbf{x}_T^k \right\|_F^2 \\ &= \| \mathbf{E}_k - \mathbf{d}_k \mathbf{x}_T^k \|_F^2 \end{aligned} \tag{2.13}$$

In the above equation, the matrix multiplication of $\mathbf{D}$ and $\mathbf{X}$ has been broken down into the sum of K rank-1 matrices. Out of which, K-1 terms are kept fixed and k^{th} term is analyzed. The matrix term $\mathbf{E}_k$ in (2.13) is the total error of the N signals except from the k^{th} signal.

The $\mathbf{E}_k$ calculated in (2.13) is restricted using the ω_k. For this we define a matrix $\mathbf{\Omega}_k$ with a size of N x $|\omega_k|$. The matrix $\mathbf{\Omega}_k$ has Ones in $(\omega_k(i), i)$ and zeroes in all other place. The matrix $\mathbf{\Omega}_k$ is multiplied to $\mathbf{x}_T^k$, which will result in a row vector $\mathbf{x}_R^k$ of dimension $|\omega_k|$. The other values are discarded due to the presence of zeroes in the $\mathbf{\Omega}_k$ matrix. Similarly, when $\mathbf{\Omega}_k$ is multiplied to

$\mathbf{E}_k$, it will restrict the overall error to error caused by the signals that use the atom of the dictionary in question [1].

The error term, therefore, can be rewritten as [1]

$$\| \mathbf{E}_k \mathbf{\Omega}_k - \mathbf{d}_k \mathbf{x}_T^k \mathbf{\Omega}_k \|_F^2 \quad = \quad \| \mathbf{E}_k^R - \mathbf{d}_k \mathbf{x}_R^k \|_F^2 \tag{2.14}$$

Now, the restricted error $\mathbf{E}_k^R$ is factorized into $\mathbf{U},\mathbf{\Delta},\mathbf{V}^T$ using SVD. The atom $\mathbf{d}_k$ is updated by the first column of $\mathbf{U}$ and the $\mathbf{x}_k^R$ is updated by the multiplication of first column of $\mathbf{V}$ and $\mathbf{\Delta(1,1)}$.

It should be noted that the solution necessarily has two of the following things. First, the columns of **D** are still normalized. Second, the support for all representations either remains constant or decreases due to potential term nulling [1],[25].

2.4 SUMMARY

This chapter deals with the literature survey of the K-SVD algorithm for dictionary learning. The K-SVD algorithm essentially has 2 stages. The first stage is sparse coding stage and the second is dictionary update stage. The sparse coding stage is done using five different algorithms viz., OMP, Batch OMP, POMP, OLS and LAOLS. These algorithms have also been discussed in this chapter.

CHAPTER 3

DISCRIMINATIVE K-SVD ALGORITHM BASED DICTIONARY LEARNING

- INTRODUCTION
- ABOUT DISCRIMINATIVE K-SVD
- DISCRIMINATIVE K-SVD ALGORITHM
- SUMMARY

3.1. INTRODUCTION

The literature survey on Discriminative K-SVD is presented in the chapter. D-KSVD is K-SVD based dictionary learning algorithm which helps us in learning dictionary with not only representational prowess but also better discriminative power which enables us to use the learned dictionary for classification problem. The addition of discriminative power is discussed in this chapter.

3.2. ABOUT DISCRIMINATIVE K-SVD

D-KSVD (Discriminative K-SVD) is a modification of the K-SVD algorithm which is used for dictionary learning. From the discussion carried out in Chapter 2 it is known that the K-SVD algorithm is one of the most popular dictionary learning algorithms used for learning the dictionary of basis functions from a set of training signals.

D-KSVD finds its application in face recognition [29], pattern recognition [32], image processing [30].

Using D-KSVD, a discriminative information is added into the dictionary learning process carried out by the basic K-SVD algorithm [29]. The use of class-specific information or labels throughout the dictionary learning process is referred to as "discriminative". The discriminative term is typically based on some form of supervised or semi-supervised learning, where the training signals are labeled with class information. For instance, in face recognition tasks, the distinction between the reconstructed face images and the corresponding class-specific prototypes might serve as the basis for the discriminative term. This is done to achieve a dictionary learning technique where a dictionary learned is effective for classification or recognition tasks in addition to sparsity. As mentioned above, this is done by adding a discriminative term [29].

The basic problem solved using K-SVD is given by [1]:

$$\min_{\mathbf{D},\mathbf{X}} \{\| \mathbf{Y} - \mathbf{DX} \|_F^2\} \textit{ subject to } \forall_i, \ \| \mathbf{x}_i \|_0 \le T_0 \tag{3.1}$$

Here, (3.1) deals with reconstruction error and sparsity only. Therefore, learned dictionary does not give an optimal result for classification work. To improve the discriminative power so that the learned dictionary gives better result, a discriminative term is used in [29],[30],[31]. The term is given by:

$$\arg\min_{\boldsymbol{\theta}} \sum_i C(\mathbf{h}_i f(\mathbf{x}_i, \boldsymbol{\theta})) + \lambda_1 \| \boldsymbol{\theta} \|_2 \tag{3.2}$$

where,

$\boldsymbol{\theta}$ = parameter of the classifier

$\mathbf{h}_i$ = label

$C(x) = \log(1+e^{-x})$ = logistic loss function

Projected gradient descent is used to find the solution to this problem since the problem we obtain by adding an extra term is very complex and cannot be solved using direct method.

A different approach can also be executed for the representation of the classifier, as given in [32,][29]. The formulation of the classifier problem is given below.

$$\arg\min_{\mathbf{W},\mathbf{b}} \| \mathbf{H} - \mathbf{W}\mathbf{x} - \mathbf{b} \|_2 + \beta' \| \mathbf{W} \|_2 \tag{3.3}$$

where, $\mathbf{W},\mathbf{b}$ = parameter for linear classifier

$\mathbf{H} = \mathbf{W}\mathbf{x} + \mathbf{b}$

$\| \mathbf{H} - \mathbf{W}\mathbf{x} + \mathbf{b} \|_2$ = Classification error

$\| \mathbf{W} \|_2$ = regularization penalty term.

It can be noted that each column of **H** is a vector and the position of the non-zero coefficient indicates the class and also **b** was set to a zero vector for simplicity.

From (3.1) and (3.3), we can formulate a problem which will learn a dictionary which has discriminative power as well as representative power. The problem is given below [29]:

$$\arg\min_{\mathbf{D},\mathbf{W},\mathbf{x}} \| \mathbf{Y} - \mathbf{D}\mathbf{x} \|_2 + \gamma \| \mathbf{H} - \mathbf{W}\mathbf{x} \|_2 + \beta \| \mathbf{W} \|_2 \text{ subject to } \forall_i, \| \mathbf{x}_i \|_0 \leq T_0 \tag{3.4}$$

Where, $\mathbf{Y}$ = Input signals
$\mathbf{D}$ = Dictionary
$\mathbf{x}$ = sparse coefficient
$\mathbf{H}$ = Label
$\mathbf{W}$ = classifier

In (3.4), γ and β are constants which define the contributions of respective terms for dictionary learning.

In [29],[32], baseline algorithm is used to solve the problem given by (3.4). The algorithm that we shall mention is a slight modification of the algorithm mentioned in [32]. Unlike [32], we consider only labeled data in this algorithm.

Step 1: Initialize $\mathbf{D}$ and $\mathbf{x}$ using K-SVD by solving

$$\min_{\mathbf{D},\mathbf{X}} \{\| \mathbf{Y} - \mathbf{D}\mathbf{X} \|_F^2\} \text{ subject to } \forall_i, \ \| \mathbf{x}_i \|_0 \leq T_0$$

Step 2: Calculate $\mathbf{W}$ using equation given below and by keeping $\mathbf{D}$ and $\mathbf{x}$ fixed.

$$\arg\min_{\mathbf{W},\mathbf{b}} \| \mathbf{H} - \mathbf{W}\mathbf{x} - \mathbf{b} \|_2 + \beta' \| \mathbf{W} \|_2$$

Step 3: Calculate $\mathbf{x}$ keeping $\mathbf{D}$ and $\mathbf{W}$ fixed.

Step 4: Calculate $\mathbf{D}$ keeping $\mathbf{x}$ and $\mathbf{W}$ fixed.

Continue step 2 to step 4 until the stopping criterion are met.

Fig 3.1. Baseline Algorithm [29].

The Baseline Algorithm mentioned above only finds the approximate solution for problem mentioned by (3.1). The baseline algorithm finds solution for sub-problem of (3.4) in each step. Although the solution computed by the baseline algorithm converges to a proper solution, it may sometimes get stuck on local minima of a sub-problem. Another notable problem of baseline algorithm is that the defined problem has three sub-problems which increases the computational burden and the convergence may be slow.

In order to overcome this problem, Discriminative K-SVD (D-KSVD) algorithm was proposed in [29],[33]. D-KSVD uses K-SVD to find the solution for all the sub-problems simultaneously. To solve the problem stated in (3.4) using D-KSVD, the problem is modified as follows.

$$\underset{\mathbf{D},\mathbf{W},\mathbf{x}}{\arg\min} \left\| \begin{pmatrix} \mathbf{Y} \\ \sqrt{\gamma}\mathbf{H} \end{pmatrix} - \begin{pmatrix} \mathbf{D} \\ \sqrt{\gamma}\mathbf{W} \end{pmatrix} \mathbf{x} \right\|_2 + \| \mathbf{W} \|_2 \text{ subject to } \forall_i, \| \mathbf{x}_i \| \leq T_0 \tag{3.5}$$

The flow chart for the dynamics within D-KSVD is given below [61]:

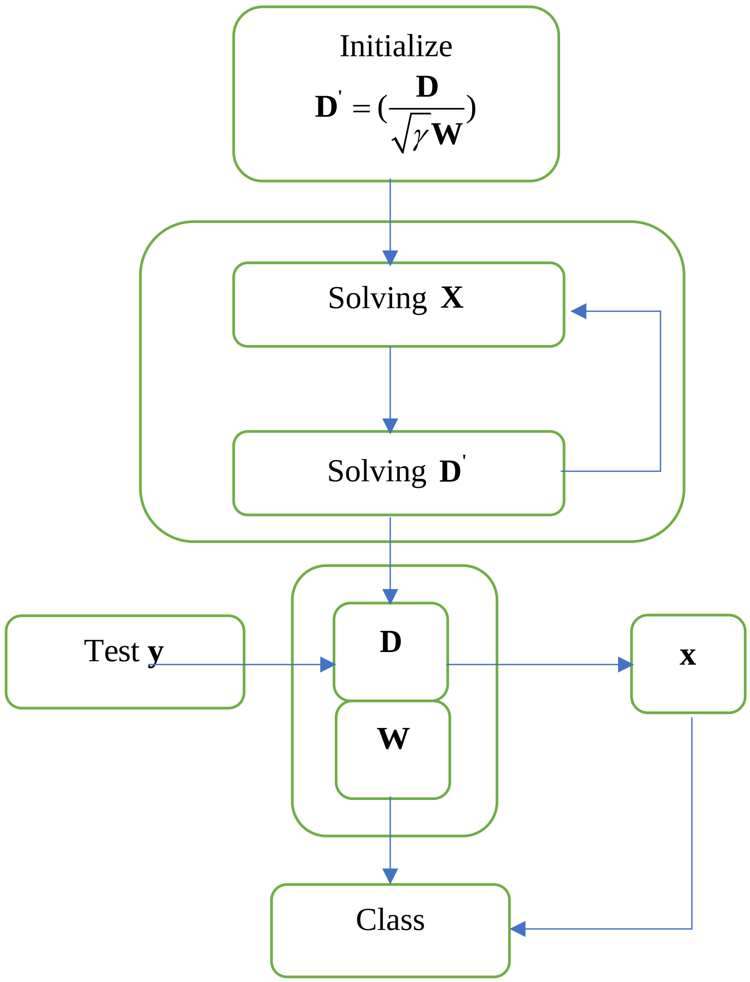

Fig 3.2. D-KSVD flowchart for classification problem [61].

It can be noted that like in K-SVD algorithm, the dictionary is normalized column-wise, the term $\begin{pmatrix} \mathbf{D} \\ \sqrt{\gamma}\mathbf{W} \end{pmatrix}$ is also normalized column-wise. Hence the term $\| \mathbf{W} \|_2$ can be neglected. Hence the problem results as follows which will be solved using K-SVD.

$$\underset{\mathbf{D},\mathbf{W},\mathbf{x}}{\arg\min} \left\| \begin{pmatrix} \mathbf{Y} \\ \sqrt{\gamma}\mathbf{H} \end{pmatrix} - \begin{pmatrix} \mathbf{D} \\ \sqrt{\gamma}\mathbf{W} \end{pmatrix} \mathbf{x} \right\|_2 \text{ subject to } \forall_i, \|\mathbf{x}_i\| \le T_0 \tag{3.6}$$

The problem stated in (3.6) can be solved by updating the atoms of the dictionary as follows

$$\underset{\mathbf{d}_k,\mathbf{x}_k}{\arg\min} \|\mathbf{E}_k - \mathbf{d}_k\mathbf{x}_k\|_F \tag{3.7}$$

$$\text{where, } \mathbf{E}_k = \mathbf{Y} - \sum_{i \ne k} \mathbf{d}_i \mathbf{x}_i$$

$\mathbf{E}_k$ is factorized into $\mathbf{U},\mathbf{\Delta},\mathbf{V}^T$ using SVD. The atom $\mathbf{d}_k$ is updated by the first column of $\mathbf{U}$ and the $\mathbf{x}_k$ is updated by the multiplication of first column of $\mathbf{V}$ and $\mathbf{\Delta(1,1)}$.

3.3. D-KSVD ALGORITHM

The detailed D-KSVD algorithm to solve the problem given in (3.6) is given below [33]:

Input: $\mathbf{Y} \in \Re^{n\times N}, \mathbf{H} \in \Re^{m\times N}, \gamma, T_0$

Output: $\mathbf{D} \in \Re^{n\times K}, \mathbf{W} \in \Re^{m\times K}, \mathbf{X} \in \Re^{K\times N}$

Step 1: Initialize

- Compute $\mathbf{D}^{(0)}$ using an initialization scheme of choice, e.g., by concatenating class specific dictionaries found with K-SVD
- Compute $\mathbf{X}^{(0)}$ for $\mathbf{Y}$ and $\mathbf{D}^{(0)}$ using sparse coding.
- Compute $\mathbf{W}^{(0)}$ using the equation given below for $\lambda = 1$

$$\mathbf{W}^{(0)} = \mathbf{H}\mathbf{z}^{(0)}$$

$$\text{Where } \mathbf{z}^{(0)} = \mathbf{X}^{(0)^T}(\mathbf{X}^{(0)}\mathbf{X}^{(0)^T} + \mathbf{I})^{-1}$$

Step 2: K-SVD

- for $k = 1$: number of iterations
- Solve the equation below using K-SVD

$$\underset{\mathbf{D},\mathbf{W},\mathbf{x}}{\arg\min} \left\| \begin{pmatrix} \mathbf{Y} \\ \sqrt{\gamma}\mathbf{H} \end{pmatrix} - \begin{pmatrix} \mathbf{D} \\ \sqrt{\gamma}\mathbf{W} \end{pmatrix} \mathbf{x} \right\|_2 \text{ subject to } \forall_i, \|\mathbf{x}_i\| \le T_0$$

- Use $(\mathbf{D}^{(0)^T}, \sqrt{\gamma}\mathbf{W}^{(0)^T})^T$
- end for

Step 3: Normalize

- $\mathbf{D} \leftarrow \{\frac{\mathbf{d}_1}{\|\mathbf{d}_1\|_2}, \frac{\mathbf{d}_2}{\|\mathbf{d}_2\|_2}, \ldots, \frac{\mathbf{d}_K}{\|\mathbf{d}_K\|_2}\}$
- $\mathbf{W} \leftarrow \{\frac{\mathbf{w}_1}{\|\mathbf{d}_1\|_2}, \frac{\mathbf{w}_2}{\|\mathbf{d}_2\|_2}, \ldots, \frac{\mathbf{w}_K}{\|\mathbf{d}_K\|_2}\}$

Fig 3.2. Discriminative K-SVD [3]

In the above-mentioned algorithm, we have $\mathbf{Y}, \mathbf{H}, \gamma$ and T_0 as input arguments, where, $\mathbf{Y} =$ Training Signals, $\mathbf{H} =$ Label matrix (Matrix with class information), $\gamma =$ Scalar that controls the relative contribution of the discriminative term and $T_o =$ Sparsity level. First, we calculate $\mathbf{X}$ keeping $\mathbf{Y}$ and $\mathbf{D}$ fixed using sparse coding. In this paper we have used OMP (Orthogonal Matching Pursuit), Batch OMP (Batch Orthogonal Matching Pursuit), POMP (Projection Based Orthogonal Matching Pursuit), OLS (Orthogonal Least Square) and LAOLS (Look Ahead Orthogonal Least Square) for sparse coding. The sparse coding has already been discussed in Chapter 2.

Next, we calculate $\mathbf{W}$ using the equation $\mathbf{W}^{(0)} = \mathbf{H}\mathbf{X}^{(0)^T}(\mathbf{X}^{(0)}\mathbf{X}^{(0)^T} + \mathbf{I})^{-1}$. $\mathbf{W}$ is a linear classifier. The linear classifier is computed and later learned to incorporate class information into the dictionary learning process. A linear classifier, such as logistic regression or support vector machine (SVM), is frequently used in D-KSVD. The classifier can be used to predict the class labels of new and unseen data after the dictionary and classifier have been trained.

Now we use K-SVD to solve (3.6). In this step we learn Dictionary $(\mathbf{D})$, Linear classifier $(\mathbf{W})$ and Sparse matrix $(\mathbf{X})$ simultaneously. These three are the sub-problems of the problem mentioned in (3.6).

The dictionary $\mathbf{D}$ and the classifier $\mathbf{W}$ that we learned from the previous step cannot be used for sparse-coding based representation as it is, since the $\mathbf{D}$ and $\mathbf{W}$ learned is normalized jointly, i.e.,

$$\left\| \begin{pmatrix} \mathbf{d}_i \\ \sqrt{\gamma}\mathbf{w}_i \end{pmatrix} \right\|_2 = 1 \tag{3.8}$$

The classifier $\mathbf{W}$ we obtained in previous stage was calculated using the dictionary which was not normalized. For this reason, the dictionary $\mathbf{D}$ cannot be readily be re-normalized. Therefore, in order to use the learned dictionary $\mathbf{D}$ and classifier $\mathbf{W}$ for sparse coding representation, they must be re-normalized. A method mentioned in [29] to obtain desired dictionary $\mathbf{D'}$ and classifier $\mathbf{W'}$ from the learned dictionary $\mathbf{D}$ and classifier $\mathbf{W}$ is given below.

$$\begin{aligned}\mathbf{D}' &= \{\mathbf{d}_1', \mathbf{d}_2', ..., \mathbf{d}_k'\} \\ &= \{\frac{\mathbf{d}_1}{\|\mathbf{d}_1\|_2}, \frac{\mathbf{d}_2}{\|\mathbf{d}_2\|_2}, ..., \frac{\mathbf{d}_k}{\|\mathbf{d}_k\|_2}\}\end{aligned} \tag{3.9}$$

$$\begin{aligned}\mathbf{W}' &= \{\mathbf{w}_1', \mathbf{w}_2', ..., \mathbf{w}_k'\} \\ &= \{\frac{\mathbf{w}_1}{\|\mathbf{d}_1\|_2}, \frac{\mathbf{w}_2}{\|\mathbf{d}_2\|_2}, ..., \frac{\mathbf{w}_k}{\|\mathbf{d}_k\|_2}\}\end{aligned} \tag{3.10}$$

The desired dictionary $\mathbf{D}'$ and classifier $\mathbf{W}'$ obtained from (3.9) and (3.10) then can be used for sparse-coding based representation.

3.4. SUMMARY

This chapter gives us the report on Discriminative K-SVD which is a K-SVD based dictionary learning algorithm. There is a detailed algorithm of D-KSVD. This algorithm will be implemented on 3 sets of different data to solve a bi class classification. The sparse coding part for the D-KSVD algorithm will be done by five different sparse coding techniques which have been discussed in Chapter 2.

CHAPTER 4

APPROXIMATE K-SVD

- INTRODUCTION
- ABOUT APPROXIMATE K-SVD
- APPROXIMATE K-SVD ALGORITHM
- SUMMARY

4.1. INTRODUCTION

This chapter shall cover the literature survey of Approximate K-SVD algorithm. This chapter also gives the detailed Approximate K-SVD algorithm. Approximate K-SVD dictionary learning algorithm requires much lesser computational burden as compared to the K-SVD dictionary learning algorithm. The reason for the same has been discussed in this chapter.

4.2. ABOUT APPROXIMATE K-SVD ALGORITHM

Approximate K-SVD algorithm is a variant of K-SVD algorithm which tries to reduce the computational burden that is required to solve a problem using original K-SVD algorithm. Once again, the basic minimization problem that we aim to solve is given by [1]:

$$\min_{\mathbf{D},\mathbf{X}} \{\| \mathbf{Y} - \mathbf{D}\mathbf{X} \|_F^2\} \text{ } \textit{subject to } \forall_i, \ \| \mathbf{x}_i \|_0 \leq T_0 \tag{4.1}$$

The problem given in (4.1) is solved by using K-SVD algorithm discussed in Chapter 2 to obtain better dictionary. In Chapter 2, we updated the dictionary atom by using Singular Value Decomposition on Error matrix $\mathbf{E}$. The error matrix $\mathbf{E}$ is restricted to have the atoms of the dictionary which is used by the corresponding signal. The size of the error matrix $\mathbf{E}$ is proportional to the number of training signals. Hence to reach an exact solution will result in computational burden. Although an exact solution is not required, we definitely want the final result to converge. Even the K-SVD algorithm converges to a local minimum instead of global [3],[1]. It can be said that the aim of the K-SVD algorithm is to obtain a better dictionary rather than an optimal one. Therefore, a faster way to obtain a better dictionary can be using the Approximate K-SVD algorithm which has less computational burden [3]. The implementation of Approximate K-SVD in [3] uses a single iteration to alternately optimize the dictionary atom $\mathbf{d}$ and the coefficient row $\mathbf{g}^T$ given by the following equation.

$$\begin{aligned} \mathbf{d} &:= \frac{\mathbf{E}_g}{\| \mathbf{E}_g \|_2} \\ \mathbf{g} &:= \mathbf{E}^T \mathbf{d} \end{aligned} \tag{4.2}$$

The method mentioned above, if executed for longer iteration will ultimately given an optimum solution. However, if the process is truncated, it gives us an approximate solution which still helps us in reducing the penalty term. The experiments carried out in [3] also show that result obtained using single iteration is sufficient rather than executing full iterations.

The Approximate K-SVD algorithm saves time and memory as it does not calculate the error matrix $\mathbf{E}$. Rather it computes only the products of this matrix with vectors which has less computational burden.

4.3. APPROXIMATE K-SVD ALGORITHM

The detailed version of the Approximate K-SVD algorithm to solve problem described by (4.1) is given below [3]:

Input: Training Signal $\mathbf{Y}$, initial dictionary $\mathbf{D_0}$, Sparsity T_0, number of iterations k

Output: Dictionary $\mathbf{D}$, sparse matrix $\mathbf{X}$ such that $\mathbf{Y} \approx \mathbf{DX}$

Step 1: Initialization: Set $\mathbf{D} := \mathbf{D_0}$

Step 2: for $n = 1...k$ do

Step 3: $\forall i : \mathbf{x}_i := \arg\min_{\mathbf{x}} \| \mathbf{y}_i - \mathbf{D}\mathbf{x}_i \|_2^2$ subject to $\| \mathbf{x}_i \| \le T_0$

Step 4: for $j = 1...L$ do

Step 5: $\mathbf{D}_j := \mathbf{0}$

Step 6: $I :=$ {indices of the signals in $\mathbf{Y}$ whose representation uses $\mathbf{d}_j$}

Step 7: $\mathbf{g} := \mathbf{X}_{j,I}^T$

Step 8: $\mathbf{d} := \mathbf{Y}_{I,g} - \mathbf{D}\mathbf{X}_{I,g}$

Step 9: $\mathbf{d} := \dfrac{\mathbf{d}}{\| \mathbf{d} \|_2}$

Step 10: $\mathbf{g} := \mathbf{Y}_I^T \mathbf{d} - (\mathbf{D}\mathbf{X}_I)^T \mathbf{d}$

Step 11: $\mathbf{D}_j := \mathbf{d}$

Step 12: $\mathbf{X}_{j,I} := \mathbf{g}^T$

Step 13: end for Step 14: end for

Fig 4.1. Approximate K-SVD Algorithm [3].

In this algorithm, first we initialize a dictionary. Next, we calculate sparse matrix using sparse coding on the present dictionary. For solving the Sparse Coding stage, the sparse coding algorithms that have been used in this thesis are OMP (Orthogonal Matching Pursuit), Batch OMP (Batch Orthogonal Matching Pursuit), POMP (Projection Based Orthogonal Matching Pursuit), OLS (Orthogonal Least Square) and LAOLS (Look Ahead Orthogonal Least Square). All these algorithms have been discussed in detail in Chapter 2.

Now we select a subset of training signals which uses the particular atom of the dictionary in question and that dictionary atom is updated. In-case of K-SVD, whole dictionary is updated which increases the computational burden. However, in Approximate K-SVD, to reduce the computational burden, only a subset of atoms chosen from the dictionary is updated. Although this process gives a reasonable result, it may lead to slight decrease of accuracy in dictionary learning process.

Hence, we can conclude that Approximate K-SVD gives faster result as compared to original K-SVD with a possibility of achieving slightly less accurate dictionary learning process. However, the actual performance will depend on the problem at hand will vary with it.

4.4. SUMMARY

The detailed discussion on Approximate K-SVD was held in this chapter. The advantages and disadvantages of Approximate K-SVD dictionary learning algorithm over K-SVD dictionary learning algorithm was also discussed in this chapter. This algorithm will be later implemented on three sets of datasets viz., The Human Motion Primitives Dataset, Single Chest Mounted Accelerometer Dataset and Road and Types Classification Dataset in the upcoming chapters.

CHAPTER 5

K-SVD BASED DICTIONARY LEARNING ALGORITHMS FOR HUMAN BEHAVIOR DETECTION

- INTRODUCTION
- PROBLEM FORMULATION
- THE HUMAN MOTION PRIMITIVE DATASET
- ALGORITHM IMPLEMENTATIONS AND RESULTS
    - APPLICATION OF K-SVD ALGORITHM
    - APPLICATION OF D-KSVD ALGORITHM
    - APPLICATION OF APPROXIMATE K-SVD ALGORITHM
    - ANALYSIS
- SUMMARY

5.1 INTRODUCTION

In this chapter we shall discuss the implementation of K-SVD based dictionary learning algorithms using Human Motion Primitive Dataset. The dataset used in this chapter is elaborated in section 5.3. The K-SVD based dictionary learning algorithms implemented in this chapter are K-SVD, D-KSVD and A-KSVD. These dictionary learning algorithms are implemented in combination with five different sparse coding algorithms. These sparse coding algorithms are OMP, Batch OMP, POMP, OLS and LAOLS and has been discussed in details in Chapter 2. The result obtained from the implementation of all the above-mentioned algorithms are tabulated and discussed. The future scope regarding application of other variants of K-SVD based dictionary learning algorithm will be discussed in this chapter.

5.2. PROBLEM FORMULATION

Human activity detection problem involves detecting and tracking human movements taking data from video or image sequence. There has been an ample number of studies on human movement detection [34-36]. It has various applications such as surveillance [37], activity recognition, human computer interaction etc.

In this chapter we shall discuss activity recognition problem. The activity recognition problem has been solved here by using algorithms like K-SVD, discussed in section 2.3 of chapter 2, D-KSVD, discussed in section 3.2 of chapter 3 and A-KSVD, discussed in section 4.2 of chapter 4. Fundamentally, we solve the human movement detection problem by solving the optimization problem given by the equation in [1]:

$$\min_{\mathbf{D},\mathbf{X}} \{\|\mathbf{Y}-\mathbf{DX}\|_F^2\} \textit{ subject to } \forall_i, \ \|\mathbf{x}_i\|_0 \le T_0 \tag{5.1}$$

Fig 5.1 below shows the human motion detection procedure used in this paper. The signals used are pre acquired.

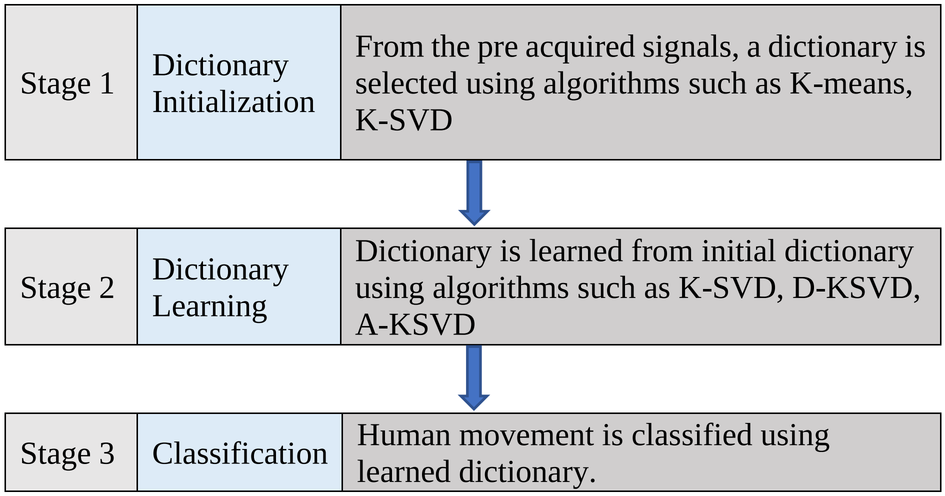

Fig 5.1. Flow chart for Human Movement Detection procedure.

In this thesis, we have primarily discussed about the bi class classification problem where one class is the event class and other is the combination of remaining other six classes. We obtain recognition rates for all the classes taking one class as an event class at a time.

5.3. THE HUMAN MOTION PRIMITIVE DATASET

The dataset used in this chapter is "Public Dataset of Accelerometer Data for Human Motion Primitives (HMP) Detection". It is a public collection of labelled accelerometer data recordings to be used for the creation and validation of acceleration models of human motion primitives. The data was taken by a tri-axial type accelerometer at 32 Hz. The accelerometer was attached to the wrist of the right hand of the test subject. The data obtained has acceleration data in 3 directions, i.e., x direction, y direction and z direction, for different human movements [58],[59],[60].

The data has 14 different types of human movement. Seven human movements have been chosen from this publicly available dataset, each having data elements more than 100. The human movements chosen are climbing stairs, drinking glass, getting up from bed, pouring water, sitting down on a chair, standing up a chair, and walking.

The total dataset created is of size (3×700), i.e., comprising sub-datasets of size (3×100) from each class, where class here refers to a particular human movement. In the thesis class 1 refers to climbing stairs, class 2 is drinking glass, class 3 is getting up from bed, class 4 is pouring water, class 5 is sitting down on a chair, class 6 is standing up a chair and class 7 is walking. At first, data from each class is divided into training data and testing data. 80% of the data is considered as training data and the rest 20% of data is considered as testing data. The initial dictionary is constructed from the training dataset by taking 80% of the total training data. The initialization process was carried out by using K-means (for K-SVD) and by using both K-means and K-SVD (for D-KSVD and A-KSVD).

The problem is solved as a bi-class classification problem. Therefore, the dictionary size is (3×128) i.e., (3×64) from each class. The testing data on the other hand is of size (3×40) i.e., (3×20) from each class. Out of (3×128) in the dictionary, (3×64) is from event class and the remaining (3×64) is combined from the remaining 6 classes. Similarly, for the testing data, out of (3×40) data, ((3×20) is from event class and the remaining (3×20) data is the combination of testing data of remaining 6 classes.

The initialized dictionary is then learned into a better dictionary using algorithms such as K-SVD (Chapter 2, section 2.3), D-KSVD (Chapter 3, section 3.2) and A-KSVD (Chapter 4, section 4.2) in combination with various sparse coding algorithms such as OMP, Batch OMP, POMP, OLS and LAOLS, which have already been discussed in section 2.3.3 of Chapter 2. The learned dictionary is then used for solving bi-class classification problems. The result of this experiment is illustrated in the subsequent sections.

5.4. ALGORITHM IMPLEMENTATION AND RESULTS

In this section we shall discuss about the results obtained from implementing various algorithms for bi class classification problem on this dataset. The algorithms in use, as mentioned before, are K-SVD, D-KSVD and A-KSVD along with sparse coding algorithms such as OMP, Batch OMP, POMP, OLS and LAOLS which have also been covered in previous chapters.

5.4.1. APPLICATION OF K-SVD ALGORITHM

Let us first discuss the application of K-SVD algorithm for dictionary learning. There are five variations of sparse coding used for dictionary learning process. All these variations have been discussed in Chapter 2 section 2.3.3. After learning the dictionary, Sparse Representation based Classification (SRC) is performed using the learned dictionary. Sparse coding used in the SRC is OMP. Parameters such as sparsity and number of iterations were varied to get the result as best as possible. However, sparsity variation for Batch OMP was not done since error based stopping criteria was used in Batch OMP.

Table 5.1. Recognition rate for bi class classification using K-SVD on HMP Dataset

Sparse coding	Recognition Rate (%)							
	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Average Recognition Rate (%)
OMP	85.00	100.00	69.99	94.01	75.67	69.00	85.73	82.07
Batch OMP	87.50	95.00	87.50	82.50	85.00	80.00	90.00	86.79
POMP	80.00	92.50	85.00	87.50	85.00	80.00	85.00	85.00
OLS	87.50	87.50	70.00	77.50	65.00	80.00	82.50	78.57
LAOLS	85.00	95.00	85.00	85.00	80.00	85.00	92.50	86.79

Table 5.1 gives the recognition rates for bi class classification using combination of K-SVD and five different sparse coding techniques. The combination of K-SVD and OMP along with other combinations has been used in [34] for learning dictionary and a combination of SRC and OMP for classification is used in the same paper. In [34] the average recognition rate for said combination was 82.07%.

In this thesis, the average recognition rate while using Batch OMP as sparse coding with K-SVD gives the average recognition rate as 86.79%. On applying POMP as sparse coding method we have an average recognition rate as 85.00%. With OLS, the average recognition rate is 78.57% which is lowest of all. Application of LAOLS gives 86.79% average recognition rate. Hence, we can conclude that, combination of K-SVD with Batch OMP or LAOLS gives the best result at 86.79% average recognition rate, which is better than the results obtained in [34], using OMP algorithm.

5.4.2. APPLICATION OF D-KSVD ALGORITHM

In this section we apply Discriminative K-SVD in combination with five different sparse coding algorithms as mentioned in previous section. For classification phase, Sparse Representation based Classification along with OMP sparse coding was used. In addition to the parameter variation done during K-SVD algorithm, we also vary γ, which is a scaler that controls the relative contribution of the discriminative term. Table 5.2 gives detailed compilation of recognition rates obtained using D-KSVD with different sparse coding.

Table 5.2. Recognition rate for bi class classification using Discriminative K-SVD on HMP Dataset

Sparse coding	Recognition Rate (%)							
	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Average Recognition Rate (%)
OMP	90.00	90.00	82.50	87.50	85.00	85.00	90.00	87.14
Batch OMP	87.50	92.50	75.00	82.50	82.50	80.00	90.00	84.29
POMP	87.50	92.50	82.50	85.00	85.00	82.50	90.00	86.43
OLS	82.50	87.50	80.00	80.00	77.50	80.00	82.50	81.43
LAOLS	85.00	90.00	87.50	87.50	80.00	80.00	90.00	85.71

As can be seen from Table 5.2, average recognition rate for bi class classification using combination of Discriminative k-SVD and OMP is 87.14%. Similarly, with D-KSVD and Batch OMP, we have average recognition rate of 84.29%. With POMP, the average recognition rate is 86.43%. The average recognition rate with D-KSVD and OLS is 81.43% and finally with LAOLS is 85.71%. Hence, while applying D-KSVD, the best result is obtained when the sparse coding stage is solved using OMP algorithm and it gave an average recognition rate of 87.14%.

5.4.3. APPLICATION OF APPROXIMATE K-SVD ALGORITHM

In a similar manner, next, Approximate K-SVD is used to learn dictionary in combination with same five sparse coding algorithms as used in section 5.3.1. Here also, the classification stage is solved using SRC with OMP as sparse coding algorithm. Table 5.3 below gives the compilation of recognition rates for various algorithms. Similar parameter variation to K-SVD algorithm is done in this algorithm as well. Computational burden in Approximate K-SVD is lower with respect to K-SVD and D-KSVD however, it trades off with its efficiency. The efficiency although is lower than the previous algorithms it gives quite acceptable results for classification problem, better than those obtained in [34].

Table 5.3. Recognition rate for bi class classification using Approximate K-SVD on HMP Dataset

Sparse coding	Recognition Rate (%)							
	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Average Recognition Rate (%)
OMP	85.00	90.00	80.00	87.50	77.50	80.00	90.00	84.29
Batch OMP	87.50	95.00	77.50	85.00	82.50	82.50	82.25	84.61
POMP	87.50	92.50	75.00	85.00	80.00	85.00	85.00	84.29
OLS	90.00	92.50	75.00	87.50	80.00	82.50	85.00	84.64
LAOLS	85.00	90.00	82.50	85.00	77.50	80.00	85.00	83.57

Applying Approximate K-SVD with OMP as sparse coding during dictionary learning process gives an average recognition rate of 84.29%. Similarly, Approximate K-SVD with Batch OMP gives an average recognition rate of 84.61%. With POMP, the average recognition rate is at 84.29%. The average recognition rate is 84.64% with OLS and 83.57% with LAOLS. We can see that we get the best average recognition rate of 84.64% with the combination of Approximate K-SVD and OLS as sparse coding process and lowest of 83.57% with Approximate K-SVD and LAOLS.

5.4.4. ANALYSIS

From the above discussion we can summarize the results as given in Table 5.4 below.

Table 5.4. Average recognition rate for different combination of K-SVD and its variants with different sparse coding techniques

Dictionary Learning Algorithm	OMP (%)	Batch OMP (%)	POMP (%)	OLS (%)	LAOLS (%)
K-SVD	86.07	86.79	85.00	78.57	86.79
DISCRIMINATIVE K-SVD	**87.14**	84.29	86.43	81.53	85.71
APPROXIMATE K-SVD	84.29	84.61	84.29	84.64	83.57

From the above table we can conclude that combination of Discriminative K-SVD and OMP gives the best result for bi class classification problem with an average recognition rate of 87.14%.

5.5. SUMMARY

In the future, this work can be continued with using different other sophisticated varies of sparse coding process in the Sparse Representation based Classification phase. Similarly, the sparse coding stage can be solved by using some other state-of-the-art varieties of K-SVD based dictionary learning algorithms. Also, in this chapter, the problems of human behavior recognition have been solved using bi-class classification only. However, solving these problems as multi class classification problems can also be explored in near future. It is expected that the recognition rates will degrade when the problems are solved as multi-class classification problems, as was observed in [34]. How to arrest this degradation in performance will pose a great challenge for these problems in near future.

CHAPTER 6

K-SVD BASED DICTIONARY LEARNING ALGORITHMS FOR HUMAN ACTIVITY RECOGNITION FROM CHEST MOUNTED ACCELEROMETER DATA

- INTRODUCTION
- PROBLEM FORMULATION
- SINGLE CHEST MOUNTED ACCELEROMETER DATASET
- ALGORITHM IMPLEMENTATION AND RESULTS
    - APPLICATION OF K-SVD ALGORITHM
    - APPLICATION OF D-KSVD ALGORITHM
    - APPLICATION OF A-KSVD ALGORITHM
    - ANALYSIS
- SUMMARY

6.1. INTRODUCTION

In this chapter, the K-SVD and its variants for dictionary learning algorithms are employed for human activity recognition problem using a Single Chest Mounted Accelerometer Dataset. The dataset used for the activity recognition purpose is elaborated in section 6.3. In this chapter, a detail description of the activity recognition performance evaluation is presented for K-SVD, D-KSVD and A-KSVD algorithms, in combination with five different sparse coding techniques mentioned before i.e., OMP, Batch OMP, POMP, OLS and LAOLS. We also discuss the future scope of applications of K-SVD based dictionary learning for this problem at hand and possible variations that can be done to improve results.

6.2. PROBLEM FORMULATION

The problem at hand is bi-class classification problem for activity recognition. We are provided with dataset which contains accelerometer data for seven different human activities. The dataset has been elaborated in section 6.3. we solve this problem by solving a minimization problem given by the equation [1]:

$$\min_{\mathbf{D},\mathbf{X}} \{\| \mathbf{Y} - \mathbf{D}\mathbf{X} \|_F^2\} \; \textit{subject to} \; \forall_i, \;\; \| \mathbf{x}_i \|_0 \leq T_0 \tag{6.1}$$

The classification problem has two stages. First, we have sparse coding stage which is done by five different sparse coding algorithms i.e., OMP, Batch OMP, POMP, OLS and LAOLS. These algorithms have been discussed in detail in Chapter 2. Second, we have dictionary update stage which is solved by using K-SVD based dictionary learning algorithm i.e., K-SVD, D-KSVD and -KSVD.

6.3. SINGLE CHEST MOUNTED ACCELEROMETER DATASET

The uncalibrated accelerometer data was collected from a wearable accelerometer mounted on the chest [48],[19]. The accelerometer data in this dataset comprises acceleration in x, y and z direction. This dataset includes 7 different activities. In our works, these activities are known as class. These activities are

i. Working at computer,

ii. Standing up, walking and going up/down the stairs
iii. Standing
iv. Walking and talking with someone
v. Going up/down stairs
vi. Walking and talking with someone
vii. Talking while standing

The dataset was collected from 15 participants. After a careful examination of this raw data available, data from six participants are chosen with relatively visible distinct features for each class. The duration of data for each signal sample was then truncated to 10 seconds to make the data uniform for chosen six participants. This discrete data was then utilized to create the data vectors of our interest using sliding temporal windows of 2 seconds with 75% of overlapping between the windows.

The resulting data obtained was of size (104×672) from all seven classes i.e., (104×96) from each class. In a similar fashion as mentioned before, the dataset was divided into training data and testing data. In this case approximately 80% of the dataset was assigned to training data i.e., size of training data is (104×546) and remaining dataset was assigned to testing data i.e., size of testing data (104×126).The problem is solved as a bi-class classification problem. Therefore, the dictionary size is (104×120) i.e., (104×60) from each class. The testing data on the other hand is of size (104×36) i.e., (104×18) from each class. Out of (104×120) in the dictionary, (104×60) is from event class and the remaining (104×60) is combined from the remaining 6 classes. Similarly, for the testing data, out of (104×36) data, (104×18) is from event class and the remaining (104×18) data is the combination of testing data of remaining 6 classes. The Dictionary initialization process was carried out by using K-means (for K-SVD) and by using both K-means and K-SVD (for D-KSVD and A-KSVD).

The initial dictionary is then learned into a better dictionary using K-SVD, D-KSVD and A-KSVD algorithms in combination with the five sparse coding algorithms chosen and mentioned in previous chapters i.e., OMP, Batch OMP, POMP, OLS and LAOLS, which have already been discussed in section 2.3.3 of Chapter 2. The learned dictionary is then used for solving bi-class classification problems. Once the dictionary is learned, in testing phase the classification problem is solved for each unknown signal using the learned dictionary, as in the previous chapter. In the classification stage, the problem is solved using SRC with OMP as sparse coding algorithm for all cases that will be discussed next. The results obtained from all these experiments are illustrated in the subsequent sections.

6.4. ALGORITHM IMPLEMENTATION AND RESULTS

In this section we shall discuss about the results obtained from implementing various algorithms for bi class classification problem on this dataset. The algorithms in use, as mentioned before, are K-SVD, D-KSVD and A-KSVD along with sparse coding algorithms such as OMP, Batch OMP, POMP, OLS and LAOLS which have also been covered in previous chapters.

6.4.1. APPLICATION OF K-SVD ALGORITHM

Application of K-SVD algorithm on the above-mentioned dataset and results obtained thus for the activity recognition problem are presented in this section. The combination of sparse coding algorithm with the K-SVD is adopted same as mentioned in chapter 5 section 5.4.1.

Table 6.1. Recognition rate for bi class classification using K-SVD on Single Chest Mounted Accelerometer dataset.

Sparse coding	Recognition Rate (%)							
	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Average Recognition Rate (%)
OMP	75.00	77.78	77.78	75.00	77.78	75.00	80.55	76.98
Batch OMP	83.33	75.00	80.55	77.78	80.55	75.00	77.78	78.57
POMP	75.00	72.22	77.78	80.55	75.00	77.78	77.78	76.59
OLS	75.00	75.00	77.78	72.22	77.78	72.22	75.00	75.00
LAOLS	86.11	77.78	80.55	77.78	80.55	77.78	80.55	**80.16**

Table 6.1 gives the Recognition rates for bi-class classification using the combination of K-SVD and five different sparse coding techniques. The average recognition rate using OMP as sparse coding with K-SVD is 76.98%. The combination of K-SVD and Batch OMP as sparse coding gives the average recognition rate of 78.57%. Using POMP with K-SVD gives average recognition rate of 76.59%. K-SVD with OLS as sparse coding gives an average recognition rate of 75.00%. The combination of K-SVD and LAOLS as sparse coding technique gives an average

recognition rate of 80.16% which is also the best average recognition rate obtained from K-SVD and combination of five different sparse coding methods.

6.4.2. APPLICATION OF D-KSVD ALGORITHM

In this section we have tabulated results obtained by applying D-KSVD in combination with five different sparse coding algorithms. The parameters such as sparsity, number of iteration and γ are varied as is described in section 5.4.2 of Chapter 5.

Table 6.2. Recognition rate for bi class classification using Discriminative K-SVD on Single Chest Mounted Accelerometer dataset.

Sparse coding	Recognition Rate (%)							
	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Average Recognition Rate (%)
OMP	80.55	83.33	77.78	80.55	77.78	80.55	80.55	80.16
Batch OMP	86.11	80.55	80.55	88.89	83.33	86.11	83.33	**84.12**
POMP	80.55	75.00	77.78	80.55	86.11	86.11	80.55	80.95
OLS	77.78	77.78	83.33	72.22	80.55	75.00	83.33	78.57
LAOLS	80.55	77.78	83.33	83.33	77.78	80.55	75.00	79.76

The average recognition rate on applying D-KSVD with OMP as sparse coding technique is80.16%. The combination of D-KSVD and Batch OMP gives an average recognition rate at 84.12%. When POMP was used as sparse coding technique, the average recognition rate was 80.95%. OLS with combination with D-KSVD gives an average recognition rate of 78.57%. And LAOLS with D-KSVD gives an average recognition rate of 79.76%. As is evident from the discussion above, the best result is produced by the combination of D-KSVD and Batch OMP as sparse coding technique where the average recognition rate is 84.12%.

6.4.3. APPLICATION OF A-KSVD ALGORITHM

Next A-KSVD algorithm is employed for the activity recognition problem in conjunction with the five chosen algorithms for obtaining sparse coding. The parameter variation adopted is similar to that of application of K-SVD which is discussed in section 5.4.1 of Chapter 5. Table 6.3

below gives recognition rates of combination of A-KSVD with five different sparse coding techniques.

Table 6.3. Recognition rate for bi class classification using Approximate K-SVD on Single Chest Mounted Accelerometer dataset.

Sparse coding	Recognition Rate (%)							
	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Average Recognition Rate (%)
OMP	80.55	80.55	77.78	83.33	86.11	83.33	83.33	82.14
Batch OMP	80.55	83.33	80.55	88.89	80.55	77.78	86.11	82.54
POMP	83.33	77.78	80.55	88.89	80.55	86.11	83.11	**82.90**
OLS	77.78	75.00	72.22	77.78	77.78	80.5	80.55	77.37
LAOLS	88.89	83.33	75.00	83.33	80.55	77.78	83.33	81.74

The average recognition rate obtained for the combination of A-KSVD and OMP is 82.14%. Batch OMP as sparse coding technique gives us an average recognition rate of 82.54%. A-KSVD and POMP gives an average recognition rate of 82.90%. The combination of A-KSVD and OLS as a sparse coding technique gives an average recognition rate of 77.37%. A-KSVD and LAOLS gives an average recognition rate of 81.74%. From the above discussion we can infer that the combination of A-KSVD and POMP gives the best average recognition rate of 82.90%.

6.4.4. ANALYSIS

From the discussions presented in previous three section, the final results obtained can be summarized in Table 6.4 given below.

Table 6.4. Average recognition rate for different combination of K-SVD and its variants with different sparse coding techniques

Dictionary Learning Algorithms	OMP (%)	Batch OMP (%)	POMP (%)	OLS (%)	LAOLS (%)
K-SVD	76.98	78.57	76.59	75.00	80.16
DISCRIMINATIVE K-SVD	80.16	**84.12**	80.95	78.57	79.76
APPROXIMATE K-SVD	82.14	82.54	82.90	77.37	81.74

From the above table we can conclude that combination of Discriminative K-SVD and Batch OMP gives the best result for bi class classification problem with an average recognition rate of 84.12%.

6.5. SUMMARY

One can easily see that the result obtained using basic K-SVD is considerably low and D-KSVD and A-KSVD algorithms were able to partially improve the performances. However, it can also be appreciated that there is definite future scope of improvement and different more sophisticated algorithms can be potentially developed to further improve the quality of performance. In the future, this work can be carried forward by introducing a feature extraction method in order to improve the recognition rate. Different variations of Sparse coding can be introduced in the Sparse Representation based Classification phase. Some other state of the art varieties of K-SVD based dictionary learning algorithms can be introduced to solve this classification problem. Multi class classification problem can also be solved on this dataset using K-SVD based dictionary learning which has been discussed in this thesis or we can introduce other varieties of K-SVD based dictionary learning algorithms for this problem as well.

CHAPTER 7

K-SVD BASED DICTIONARY LEARNING ALGORITHMS FOR CLASSIFICATION OF ROAD AND TYPES

- INTRODUCTION
- PROBLEM FORMULATION
- ROAD AND TYPES CLASSIFICATION DATASET
- ALGORITHM IMPLEMENTATION AND RESULTS
    - APPLICATION OF K-SVD ALGORITHM
    - APPLICATION OF D-KSVD ALGORITHM
    - APPLICATION OF A-KSVD ALGORITHM
    - ANALYSIS
- SUMMARY

7.1. INTRODUCTION

In this chapter, we shall discuss the third problem that has been solved in this thesis using K-SVD based dictionary learning, based on a dataset acquired using accelerometer data. The problem under consideration is classification of road and type using JINS-MEME smart glasses [55]. These smart glasses are developed using three-point EOG and six-axis inertial measurement unit (IMU) equipped with an accelerometer and a gyroscope [51]. Detailed descriptions about the dataset will be presented in the upcoming sections. Here also, K-SVD, D-KSVD, and A-KSVD based dictionary learning algorithms have been implemented for the classification problem at hand and the results are obtained when these DL algorithms are implemented in conjunction with previously mentioned five sparse coding algorithms. The chapter will also discuss the future scope in the context of the specific problem considered in this chapter.

7.2. PROBLEM FORMULATION

The problem remains the same as is mentioned in Chapter 6, Section 6.2. Here also we solve the minimization problem given by [1]:

$$\min_{\mathbf{D},\mathbf{X}} \{\| \mathbf{Y} - \mathbf{D}\mathbf{X} \|_F^2\} \text{ } \textit{subject to } \forall_i, \ \| \mathbf{x}_i \|_0 \leq T_0 \tag{7.1}$$

K-SVD based dictionary learning algorithms and five different sparse coding algorithms which have been discussed in the previous chapters are implemented to obtain desired solution for classification problem.

7.3. ROAD AND TYPES CLASSIFICATION DATASET

As mentioned before, JINS MEME Smart Glasses were used to acquire the dataset and the JINS MEME Smart Glasses is equipped with three-point EOG and six axes inertial measurement unit (IMU) with an accelerometer and gyroscope. The sampling frequency of the data acquired is 100 Hz. The data were acquired from 30 subjects under real driving conditions. Out of 30 subjects, 20 were experienced drivers and the remaining 10 subjects were students attending the driving school. The data were labelled during the drive and broadly divided into 4 groups according to the types of roads [51]. The different groups (or Classes) considered are given as follows

i. Highway
ii. City road
iii. Underdeveloped area
iv. Housing estate

In this thesis, out of 30 subjects, 10 subjects were chosen at random. 50 signals were chosen from each subject. For each signal under consideration there were ten data recorded i.e. (ACC_X, ACC_Y, ACC_Z, GYRO_X, GYRO_Y, GYRO_Z, EOG_L, EOG_R, EOG_H, EOG_V). ACC_X, ACC_Y and ACC_Z are data recorded from an accelerometer each corresponding to x-axis, y-axis and z- axis respectively. Similarly, GYRO_X, GYRO_Y and GYRO_Z are data recorded from a gyroscope each corresponding to x-axis, y-axis and z-axis respectively. Using electrodes attached to the skin around the eyes, EOG measures changes in the electric potential field caused by eye movements. By analyzing these changes, it's possible to track relative eye movements i.e., how a person is looking at something. EOG_R means measurements of eye movement in right position, EOG_L means measurements of eye movement in the right position and EOG_H means measurements of eye movement horizontally and EOG_V means measurements of eye movement vertically. This way for one class the data acquired or chosen was of size (10×500). Therefore, considering all four classes, the total size of the dataset was ((10×2000). It means four such sets of data were created from the actual data available, one each for each class. The dataset was then divided into training data and testing data. 80% of the dataset was assigned as training data, i.e., the size of the training data was (10×1600). Remaining 20% i.e., (10×400) data were assigned as testing data.

The problem is solved as a bi-class classification problem. Therefore, the dictionary size is (10×800) i.e., (10×400) from each class. The testing data on the other hand is of size (10×200) i.e., (10×100) from each class. Out of (10×800) in the dictionary, (10×400) is from event class and the remaining (10×400) is combined from the remaining 3 classes. Similarly, for the testing data, out of (10×200) data, (10×100) is from event class and the remaining (10×100) data is the combination of testing data of remaining 3 classes. The Dictionary initialization process was carried out by using K-means (for K-SVD) and by using both K-means and K-SVD (for D-KSVD and A-KSVD).

Like in the previous two chapters, the initial dictionary is then learned into a better dictionary using K-SVD, D-KSVD and A-KSVD based dictionary learning algorithms and using OMP, Batch OMP, POMP, OLS and LAOLS algorithms, in each DL algorithm, to solve the sparse coding stage. learned dictionary is then used for solving bi-class classification problems. The classification problem is solved using SRC with OMP as sparse coding algorithm for all the application that will be discussed below. The result of this experiment is illustrated in the subsequent sections.

7.4. ALGORITHM IMPLEMENTATION AND RESULTS

In this section we shall discuss about the implementation of K-SVD based dictionary learning algorithms (K-SVD, D-KSVD and A-KSVD) in combination with five different sparse coding techniques. The results thus obtained is tabulated and explained in detail.

7.4.1. APPLICATION OF K-SVD ALGORITHM

Implementation of K-SVD combined with five different sparse coding techniques viz., OMP, Batch OMP, POMP, OLS and LAOLS is discussed in this section. Parameter variation includes variation of sparsity and number of iterations.

Table 7.1. Recognition rate for bi class classification using K-SVD on Road and Types Classification Dataset

Sparse coding	Recognition Rate (%)				
	Class 1	Class 2	Class 3	Class 4	Average Recognition Rate (%)
OMP	75.00	76.00	92.50	93.50	**84.25**
Batch OMP	74.00	75.50	91.00	79.00	79.88
POMP	84.50	76.50	84.50	76.50	80.50
OLS	90.00	76.00	94.50	72.00	83.13
LAOLS	83.00	82.50	82.50	80.00	82.00

The average recognition rate when K-SVD is implemented with OMP is 84.25%. The combination of K-SVD and Batch OMP gives an average recognition rate of 79.88%. When used with POMP, it gives an average recognition rate of 80.50%. The combination of K-SVD and OLS gives an average recognition rate of 83.13% and with LAOLS, it gives an average recognition rate of 82.00%. From the discussion, we can say that combination of K-SVD and OMP as a sparse coding technique gives the best result at 84.25% average recognition rate.

7.4.2. APPLICATION OF D-KSVD ALGORITHM

In this section, we implement D-KSVD in combination with five different sparse coding techniques mentioned in section 7.4.1. Parameter such as sparsity, number of iteration and γ are varied as is mentioned in section 5.4.2 of chapter 5.

Table 5.2. Recognition rate for bi class classification using Discriminative K-SVD on Road and Types Classification Dataset.

Sparse coding	Recognition Rate (%)				
	Class 1	Class 2	Class 3	Class 4	Average Recognition Rate (%)
OMP	93.00	76.50	88.50	84.50	85.63
Batch OMP	92.50	75.50	71.00	69.00	77.00
POMP	93.50	80.00	97.00	80.00	**87.63**
OLS	78.50	71.00	79.50	75.50	76.13
LAOLS	89.50	84.00	86.00	80.50	85.00

The average recognition rate of 85.63% is obtained when D-KSVD is implemented along with OMP. Similarly, D-KSVD in combination with Batch OMP gives an average recognition rate of 77.00%. POMP and D-KSVD produces an average recognition rate of 87.63%. the combination of OLS and D-KSVD gives an average recognition rate of 76.13%. When LAOLS is used with D-KSVD, the recognition rate obtained was 85.00%. From the discussion, it is evident that the combination of D-KSVD and POMP as a sparse coding technique gives the best average recognition rate of 87.63%.

7.4.3. APPLICATION OF A-KSVD ALGORITHM

In this section we shall discuss the implementation of A-KSVD combined with five different sparse coding techniques mentioned in section 7.4.1.

Table 7.3. Recognition rate for bi class classification using Approximate K-SVD on Road and Types Classification Dataset.

Sparse coding	Recognition Rate (%)				
	Class 1	Class 2	Class 3	Class 4	Average Recognition Rate (%)
OMP	83.00	83.50	84.00	81.00	**82.88**
Batch OMP	88.50	73.00	80.00	79.50	80.25
POMP	88.50	75.00	87.00	79.50	82.50
OLS	76.50	73.50	82.50	79.50	78.00
LAOLS	85.50	61.50	95.00	80.50	80.63

The average recognition rate of A-KSVD when combined with OMP is 82.88%. With Batch OMP, A-KSVD gives an average recognition rate of 80.25%. A-KSVD with POMP gives an average recognition rate of 82.50%. The average recognition rate with OLS is 78.00% and with LAOLS is 80.63%. The best average recognition rate is 82.88% given by the combination of A-KSVD and OMP.

7.4.4. ANALYSIS

From the discussions presented in previous three section, the final results obtained can be summarized in Table 7.4 given below.

Table 7.4. Average recognition rate for different combination of K-SVD and its variants with different sparse coding techniques

Dictionary Learning Algorithms	OMP (%)	Batch OMP (%)	**POMP** (%)	OLS (%)	LAOLS (%)
K-SVD	84.25	79.88	80.50	83.13	82.00
DISCRIMINATIVE K-SVD	85.63	77.00	**87.63**	76.13	85.00
APPROXIMATE K-SVD	82.88	80.25	82.50	78.00	80.63

From the table it is evident that combination of D-KSVD and POMP gives the best recognition rate of 87.63% amongst all the variation discussed.

7.5. SUMMARY

The work can be continued in the future by introducing other variants of K-SVD and Sparse Coding Algorithms. We can increase the number of training signal and work on it. Variation in sparse coding in the Sparse Representation based Classification problem can be introduced. Multi-class classification problem can also be solved by using K-SVD, D-KSVD and A-KSVD along with other variants of K-SVD based Dictionary Learning algorithms.

CHAPTER 8
CONCLUSION

- SYNOPSIS
- FUTURE SCOPE

8.1. SYNOPSIS

The most important aim of the thesis is the understanding of the K-SVD based dictionary learning algorithm. These algorithms are used to learn a dictionary which is then used for Sparse Representation based Classification problem. The algorithm used to learn dictionary in this thesis are K-SVD, D-KSVD and A-KSVD. The dictionary learning stage goes hand in hand with the sparse coding stage. So, it is equally important to understand the Sparse Coding algorithms. There are five different types of sparse coding algorithms that are used in this thesis.

Chapter 1 gives the background and the motivation for Dictionary Learning. In Chapter 2 we have literature survey about K-SVD algorithm and five different sparse coding algorithms. They are OMP, Batch OMP, POMP, OLS and LAOLS. Chapter 3 contains detailed literature on D-KSVD algorithm. A-KSVD algorithm is explained in Chapter 4. Chapter 5 gives us application of K-SVD based dictionary learning algorithms in combination with five different sparse coding algorithms on Human Motion Primitives Dataset. The results of these application are tabulated and explained in the same chapter. Similarly, Chapter 6 gives the application of K-SVD based dictionary learning algorithms in combination with five different sparse coding algorithms on Single Chest Mounted Accelerometer Dataset. It has been established from the performance analysis that the Discriminative K-SVD based approach can be effectively employed for recognition of human motion primitives. Chapter 7 gives the application of K-SVD based dictionary learning algorithms in combination with five different sparse coding algorithms on Dataset for Classification of Road and Types using EOG Smart Glasses. The data has been acquired using accelerometer, gyroscope and EOG (Electrooculography) Smart Glasses. The performance analysis of K-SVD based dictionary algorithm on this dataset has been done in this chapter.

Finally, we conclude the thesis by writing a synopsis of this thesis in Chapter 8. We also discuss the few possible aspects which can be looked up on in the future. The references has been placed at the very last of this thesis.

8.2. FUTURE SCOPE

In today's world, where everything is automated, Dictionary learning plays a vital role in developing AAL, security surveillance etc. In order to smoothly perform such developments, we must delve deep into the field called dictionary learning. In this thesis, we have used K-SVD, D-KSVD and A-KSVD algorithms for dictionary learning process. However, there are many other sophisticated K-SVD based dictionary learning algorithms which can be implemented. With each K-SVD variant, different state of the art sparse coding algorithms can be implemented. Nevertheless, due to time-bounded constraints associated with a Master's Thesis, those aspects could not be covered.

This thesis can be continued in the future by introducing other sophisticated K-SVD variants such as Block K-SVD, Parallel K-SVD, Structured K-SVD etc. Block K-SVD, unlike original K-SVD, divides the dictionary into blocks and updates them block-wise thereby reducing computational burden. In Parallel K-SVD, the computational burden is reduced by distributing the computation process among multiple machines. This helps in learning dictionary much faster specially when the datasets are large. In Structured K-SVD, unlike K-SVD where all the element in the dictionary were updated independently, additional constraints are introduced which will impose certain structure to the dictionary. Implementation of these K-SVD algorithms reduces the computational burden and may even provide us with better result for classification problem.

Sparse coding during dictionary learning process was done by five different algorithms viz., OMP, Batch OMP, POMP, OLS and LAOLS. However, in the future, we can implement other state-of-the-art algorithms for sparse coding during dictionary learning process such as Least Absolute Shrinkage and Selection Operator (LASSO), Iterative Soft Thresholding (IST), FOCal Underdetermined System Solver (FOCUSS) etc.

Similarly, sparse coding in the Sparse Representation based Classification (SRC) can be done by using state-of-the-art algorithms. In this thesis, the SRC was done using OMP as the sparse coding technique. We can start with implementing sparse coding algorithms discussed in this thesis such as Batch OMP, POMP, OLS and LAOLS and move on to other sparse coding techniques such as Least Absolute Shrinkage and Selection Operator (LASSO), Iterative Soft Thresholding (IST), FOCal Underdetermined System Solver (FOCUSS) etc.

This thesis mainly focuses on bi-class classification problem. However, multi-class classification problem can be solved using same approach. However, recognition rates for multi-class classification are not satisfactory. To improve the recognition rate for such problem will be a great challenge in the near future.

[1] M. Aharon, M. Elad and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311-4322, Nov. 2006, doi: 10.1109/TSP.2006.881199.

[2] Michael Elad, *Sparse and Redundant representation. From Theory to Applications in Signal and Image Processing,* New York, NY, USA: Springer, 2010.

[3] Ron Rubinstein, Michael Zibulevsky and Michael Elad, "Efficient Implementation of the K-SVD Algorithm Using Batch Orthogonal Matching Pursuit", *CS Technion*. 40, 2008.

[4] P. Irofti and B. Dumitrescu, "Overcomplete Dictionary Design: The Impact of the Sparse Representation Algorithm," in *20th International Conference on Control Systems and Computer Science*, Bucharest, Romania, 2015, pp. 901-908, doi: 10.1109/CSCS.2015.50.

[5] S. Chen, C. F. N. Cowan and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302-309, March 1991, doi: 10.1109/72.80341.

[6] S. Kumar and G. R. Sinha, "Noise Reduction in Audio Signals using Orthogonal Least Squares", *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 6, no. 3, 2013.

[7] Jianhua Zhang and Dongxia Yuan, "Sparse Signal Denoising Based on Orthogonal Least Squares Regression", *Journal of Signal Processing Systems*, vol 86, no. 1, 2017

[8] S. Kumar and S. V. R. Srinivasan, "Adaptive Linear Prediction for Denoising and Compression of Speech Signals Using Orthogonal Least Squares", *IEEE Transactions on Audio, Speech, and Language Processing* , vol. 16, no. 7, 2008.

[9] S. Srinivas and T. V. Sreenivas, "Speech Enhancement Using Orthogonal Least Squares Algorithm for Wiener Filter Design", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, 2011.

[10] Wai-Chi Fang, Ping-Feng Pai, and Cheng-Yuan Chang, "A New Adaptive Orthogonal Least Squares Algorithm for Image Compression", *Journal of Visual Communication and Image Representation*, vol. 18, no. 5, 2007.

[11] Jianhua Zhang and Dongxia Yuan, “Sparse Signal Denoising Based on Orthogonal Least Squares Regression”, *Journal of Signal Processing Systems*, vol. 86, no. 1, 2017.

[12] Yuanyuan Zhang, Xiaoqing Wang, and Lingyun Xiang, “Feature Extraction and Classification of Human Emotions Based on Look-Ahead Orthogonal Least Squares Algorithm", *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 7, no. 3, 2014.

[13] Jianguo Lu, Fei Chen, and Xinyu Liu, “Look-Ahead Orthogonal Least Squares for Feature Extraction in Pattern Recognition”, in *Proceedings of the IEEE International Conference on Signal Processing, Communications and Computing*, 2014

[14] Weiguo Gong, Yong Ma, and Ying Li, “Look-Ahead Orthogonal Least Squares for Efficient Sparse Representation and Classification of Hyperspectral Images”, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 3, 2017.

[15] Guohui Zhang, Zhijie Yan, and Wenjuan Fan, “Speech Signal Feature Extraction Based on Look-Ahead Orthogonal Least Squares Algorithm”, in *Proceedings of the International Conference on Mechatronics and Control Engineering*, 2016

[16] Shukai Duan, Yun Liu, and Wanzhong Chen, “Sparse Coding via Orthogonal Matching Pursuit for Pattern Recognition of ECG Signals”, in *Proceedings of the IEEE International Conference on Digital Signal Processing*, 2018.

[17] Jun Wan, Kaiqi Huang, and Tieniu Tan, “Sparse Representation Based Classification using Orthogonal Matching Pursuit for Face Recognition”, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[18] Hongmei Zhang and Yanhui Guo, “Audio Signal Separation Based on Orthogonal Matching Pursuit and Nonnegative Matrix Factorization”, *Journal of Signal Processing Systems*, 2015.

[19] P. Casale, O. Pujol and P. Radeva, “Personalization and user verification in wearable systems using biometric walking patterns” Personal and Ubiquitous computing, 16(5), 563-580, 2012, https://archive.ics.uci.edu/ml/datasets/Activity+Recognition+from+Single+Chest-Mounted+Accelerometer

[20] M. Elad and M. Aharon, "Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736-3745, Dec. 2006, doi: 10.1109/TIP.2006.881969.

[21] Michael Elad and Michal Aharon, "Sparse Overcomplete Representations for Image Analysis", in *Proceedings of the IEEE*, vol. 98, no. 6,2010

[22] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng, "Sparse Representation for Signal and Image Processing: An Overview", in *Proceedings of the IEEE*, vol. 106, no. 5, 2018.

[23] Rodrigo Cabral Farias and Marcelo Porto, "KSVD-Based Audio Denoising: An Evaluation on the Demand for Real-Time Processing", *IEEE Signal Processing Letters*, vol. 24, no. 2, 2017.

[24] Seung-Hoon Lee and Young-Oh Song, "Speech Enhancement Using K-SVD Based Dictionary Learning", *IEEE Transactions on Audio, Speech, and Language Processing*, vol 21, no. 10, 2013.

[25] S. Lesage, R. Gribonval, F. Bimbot and L. Benaroya, "Learning unions of orthonormal bases with thresholded singular value decomposition," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005., Philadelphia, PA, USA, 2005, pp. v/293-v/296 vol. 5, doi: 10.1109/ICASSP.2005.1416298.

[26] A. Gersho and R. M. Gray, "Vector Quantization and Signal Compression." Norwell, MA: Kluwer Academic, 1991.

[27] S. Chen, S. Billings, and W. Luo, "Orthogonal Least Squares Methods and Their Application to Non-Linear System Identification," *Int. J. Control*, vol. 50, no. 5, pp. 1873–1896, 1989.

[28] S. Chatterjee, M. Vehkapera, and M. Skoglund, "Projection-Based and Look-Ahead Strategies for Atom Selection," *IEEE Trans. Signal Proc.*, vol. 60, no. 2, pp. 634–647, Feb. 2012.

[29] 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*, San Francisco, CA, USA, 2010, pp. 2691-2698.

[30] J. Mairal, F. Bach, J. Ponce, G. Sapiro and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 2008, pp. 1-8, doi: 10.1109/CVPR.2008.4587652.

[31] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro and Andrew Zisserman, "Supervised dictionary learning", in *Proceedings of the 21st International Conference on Neural Information Processing Systems (NIPS'08)*, Curran Associates Inc., Red Hook, NY, USA, 1033-1040, 2008

[32] D. S. Pham and S. Venkatesh, "Joint learning and dictionary construction for pattern recognition," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 2008, pp. 1-8, doi: 10.1109/CVPR.2008.4587408.

[33] I. Kviatkovsky, M. Gabel, E. Rivlin and I. Shimshoni, "On the Equivalence of the LC-KSVD and the D-KSVD Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 2, pp. 411-416, 1 Feb. 2017, doi: 10.1109/TPAMI.2016.2545661.

[34] Pubali De, Amitava Chatterjee and Anjan Rakshi, "Recognition of Human Behavior for Assisted Living Using Dictionary Learning Approach", *IEEE Sensors Journal*, vol. 18, no. 6, pp. 2434-2441.

[35] C. Debes, A. Merentitis, S. Sukhanov, M. Niessen, N. Frangiadakis, and A. Bauer, "Monitoring activities of daily living in smart homes: Understanding human behavior," *IEEE Signal Process. Mag.*, vol. 33, no. 2, pp. 81–94, Mar. 2016.

[36] B. Ando, S. Baglio, C. O. Lombardo, and V. Marletta, "RESIMA: Adaptive paradigms for the user localization in indoor environments," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (IMTC)*, Pisa, Italy, May 2015, pp. 1024–1029

[37] Navneet Dalal and Bill Triggs, "Real-Time Human Detection and Tracking for Surveillance Systems", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005

[38] K. Engan, S. O. Aase, and J. H. Husoy, "Multi-frame compression: Theory and design," *EURASIP Signal Process*., vol. 80, no. 10, pp. 2121–2140, 2000.

[39] K. Engan, S. O. Aase and J. Hakon Husoy, "Method of optimal directions for frame design," in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, Phoenix, AZ, USA, 1999, pp. 2443-2446 vol.5, doi: 10.1109/ICASSP.1999.760624.

[40] K. Engan, B. D. Rao, and K. Kreutz-Delgado, “Frame design using focuss with method of optimal directions (mod),” *Norwegian Signal Process. Symp., 1999*, vol. 65-69

[41] M. S. Lewicki and B. A. Olshausen, “A probabilistic framework for the adaptation and comparison of image codes,” *J. Opt. Soc. Amer. A: Opt., Image Sci. Vision*, vol. 16, no. 7, pp. 1587–1601, 1999.

[42] B. A. Olshausen and D. J. Field, “Natural image statistics and efficient coding,” *Network Comp. Neural Syst.*, vol. 7, no. 2, pp. 333–339, 1996.

[43] B. A. Olshausen and B. J. Field, “Sparse coding with an overcomplete basis set: A strategy employed by v1?,” *Vision Res.*, vol. 37, pp. 3311–3325, 1997.

[44] M. S. Lewicki and T. J. Sejnowski, “Learning overcomplete representations,” *Neural Comp.*, vol. 12, pp. 337–365, 2000.

[45] Pubali De, Amitava Chatterjee and Anjan Rakshit, “PIR Sensor-Based AAL Tool for Human Movement Detection: Modified MCP-Based Dictionary Learning Approach”, *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 10, OCTOBER 2020, pp. 7377-7385

[46] C. Debes, A. Merentitis, S. Sukhanov, M. Niessen, N. Frangiadakis, and A. Bauer, “Monitoring activities of daily living in smart homes: Understanding human behavior,” *IEEE Signal Process. Mag.*, vol. 33, no. 2, pp. 81–94, Mar. 2016.

[47] L. Scalise et al., “Multiparameter electromagnetic sensor for AAL indoor measurement of the respiration rate and position of a subject,” in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (IMTC)*, Pisa, Italy, May 2015, pp. 664–669.

[48] Casale, Pierluigi & Pujol, Oriol & Radeva, Petia. (2009). BeaStreamer-v0.1: a new platform for Multi-Sensors Data Acquisition in Wearable Computing Applications.

[49] J. T. Zhou, J. Du, H. Zhu, X. Peng, Y. Liu and R. S. M. Goh, “AnomalyNet: An Anomaly Detection Network for Video Surveillance,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, pp. 2537-2550, Oct. 2019, doi: 10.1109/TIFS.2019.2900907.

[50] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Rev*., vol. 43, no. 1, pp. 129–159, 2001

[51] Rafał Doniec, Natalia Piaseczna, Frédéric Li, September 4, 2022, "A Dataset for classification of road and types using EOG smart glasses", IEEE Dataport, doi: https://dx.doi.org/10.21227/4yte-5s06

[52] I. F. Gorodnitsky and B. D. Rao, “Sparse signal reconstruction from limited data using FOCUSS: A re-weighted norm minimization algorithm,” *IEEE Trans. Signal Process.*, vol. 45, pp. 600–616, 1997.

[53] B. D. Rao and K. Kreutz-Delgado, “An affine scaling methodology for best basis selection,” *IEEE Trans. Signal Process.*, vol. 47, pp. 187–200, 1999.

[54] B. D. Rao, K. Engan, S. F. Cotter, J. Palmer, and K. Kreutz-Delgado, “Subset selection in noise based on diversity measure minimization,” *IEEE Trans. Signal Process.*, vol. 51, pp. 760–770, 2003

[55] D. Needell and J.A. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples” *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, 2009, Pages 301-321,

[56] M. A. Davenport, D. Needell and M. B. Wakin, “Signal Space CoSaMP for Sparse Recovery with Redundant Dictionaries”, *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6820-6829, Oct. 2013

[57] Meng, Xiao-Li, and David van Dyk. “The EM Algorithm--An Old Folk-Song Sung to a Fast New Tune,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 59, no. 3, 1997, pp. 511–67.

[58] B. Bruno, F. Mastrogiovanni, A. Sgorbissa, T. Vernazza and R. Zaccaria, "Analysis of human behavior recognition algorithms based on acceleration data," in *2013 IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, 2013, pp. 1602-1607, doi: 10.1109/ICRA.2013.6630784.

[59] B. Bruno, F. Mastrogiovanni, A. Sgorbissa, T. Vernazza and R. Zaccaria, "Human motion modelling and recognition: A computational approach," in *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, Seoul, Korea (South), 2012, pp. 156-161, doi: 10.1109/CoASE.2012.6386410.

[60] UCI Machine Learning Repository. Dataset for ADL Recognition with Wrist-Worn Accelerometer Data Set. Accessed: Oct. 9, 2016. [Online]. Available: https://archive. ics.uci.edu/ml/datasets/Dataset+for+ADL+Recognition+ with+Wrist-worn+Accelerometer

[61] W. Liu, C. Song and Y. Wang, "Facial expression recognition based on discriminative dictionary learning," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Tsukuba, Japan, 2012, pp. 1839-1842.