Tensor decomposition of multi-channel wearable sensors for Parkinson’s disease assessment

Vignesh Ramji
vxr9819@rit.edu

Follow this and additional works at: https://scholarworks.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact rit scholarworks@rit.edu.
Tensor decomposition of multi-channel wearable sensors for Parkinson’s disease assessment

by

Vignesh Ramji

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Engineering

Supervised by

Assistant Professor Dr. Behnaz Ghoraani
Department of Biomedical Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York
Dec 2016

Approved by:

Dr. Behnaz Ghoraani, Assistant Professor
Thesis Advisor, Department of Biomedical Engineering

Dr. Eli Saber, Professor
Committee Member, Department of Electrical Engineering

Dr. Panos Markopoulos, Assistant Professor
Committee Member, Department of Electrical Engineering
Thesis Release Permission Form

Rochester Institute of Technology
Kate Gleason College of Engineering

Title:

Tensor decomposition of multi-channel wearable sensors for Parkinson’s disease assessment

I, Vignesh Ramji, hereby grant permission to the Wallace Memorial Library to reproduce my thesis in whole or part.

__________________________________________________________
Vignesh Ramji

__________________________________________________________
Date
Dedication

To my wonderful parents, Ramji Krishnan, Padma Ramji and Swathi Ramji for their unconditional sacrifice, love and encouragement in every step of my life to help me achieve my goals.
Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor Dr. Behnaz Ghoraani for her continuous support of my research and for her consistent encouragement towards my ideas and goals. I am eternally grateful to her for trusting my potential and considering me as a promising candidate to do this thesis work, which without her guidance, would not have been possible. A special thanks to Dr. Michelle Burrack who was present whenever I needed clarification regarding any patient I was dealing with. I would like to thank my committee members Dr. Panos Markopoulos and Dr. Eli Saber, for serving on my committee and for being supportive in both my research and my courses. I would also like to thank Dr. Gill Tsouri, Dr. Sohail Dianat, Dr. Ferat Sahin, Dr. Panos P. Markopoulos and Dr. Eli Saber for their teaching and/or their valuable inputs and assistance either in my coursework or in any other way. Lastly, my extended gratitude to my family, colleagues and friends for providing me with the motivation to keep working all the time.
Abstract

Tensor decomposition of multi-channel wearable sensors for Parkinson’s disease assessment

Vignesh Ramji

Supervising Professor: Dr. Behnaz Ghoraani

The cure for Parkinson’s disease is considered as one of the greatest challenges in chronic neurological disorder therapy, motivating efforts to provide information to guide therapy adjustments. This disease affects the patients day to day tasks which may vary from drinking water to a more complex task like folding laundry. Postural instability and rigidity of motion can be defined as some of the main symptoms for Parkinson’s disease. In order to better understand and analyze the patients suffering from this disease, the patients were asked to maintain records in a diary of times when they felt an unusual behavior while doing a particular task. Due to the difficulty in maintaining such records, each patient is asked to wear inertial sensors that monitor various movements of the patient. With the help of mathematical tools like Tensors, data fusion is carried out on the signal received from the sensors in order to determine the severity of Parkinson’s Disease. Using machine learning algorithms, it is possible to determine the accuracy with which the developed algorithm manages to determine the extent by which each patient is affected by the Parkinson’s disease.
# Contents

Dedication ................................................................. iv

Acknowledgments ......................................................... v

Abstract ................................................................. vi

1 Introduction .............................................................. 1
   1.1 Motivation ......................................................... 1
   1.2 Objectives ......................................................... 2
   1.3 Contribution ....................................................... 3
   1.4 Thesis Organization ............................................... 4

2 Experimental Setup and Background ................................. 5
   2.1 Test Setup ......................................................... 5
   2.2 Data sets ........................................................ 7

3 Methodology ............................................................. 9
   3.1 Matrix and Tensor Decomposition ................................. 9
      3.1.1 Fundamentals of Matrices .................................. 9
      3.1.2 Introduction to Tensors .................................... 14
      3.1.3 Tensor Unfolding ............................................. 16
      3.1.4 Tensor Multiplication ...................................... 17
      3.1.5 Tensor Decomposition ...................................... 20
   3.2 Canonical Polyadic Decomposition ............................... 22
   3.3 Low Rank Multi-linear Rank Approximation ..................... 27
   3.4 Short Time Fourier Transform (STFT) ............................ 35
   3.5 Introduction to Machine Learning ............................... 39
      3.5.1 Types of Machine Learning Algorithms ..................... 39
      3.5.2 K-means Clustering .......................................... 41
      3.5.3 Support Vector Machines .................................... 45
# Proposed Methods

4.1 Tensors Organization ................................................. 49
4.2 Tensor Decomposition ............................................. 50
4.3 Feature Selection and Extraction .................................. 52
4.4 Labeling Methodology ................................................ 55
   4.4.1 Hard Label ...................................................... 55
   4.4.2 Fuzzy Label ....................................................... 57

# Results and Analysis .................................................. 58
5.1 Short Data Set: K means - Hard Label ............................... 58
5.2 Short Data Set: K means - Fuzzy Label .............................. 59
5.3 Short Data Set: Support Vector Machine (SVM) .................. 60
5.4 Long Data Set: K means - Hard Label ............................... 61
5.5 Long Data Set: K means - Fuzzy Label .............................. 62
5.6 Long Data Set: Support Vector Machine (SVM) ................. 62

# Conclusion and Future Works ......................................... 64
6.1 Contribution of this Thesis ......................................... 64
6.2 Future Works .......................................................... 65
   6.2.1 Varying sensor combinations .................................. 65
   6.2.2 Accelerometer Signals ......................................... 65
   6.2.3 Other machine learning algorithms ........................... 66
   6.2.4 Tensor Decomposition Techniques ............................ 66

Bibliography ............................................................... 67
List of Tables

4.1 State based activity categorization ........................................... 49
4.2 Example of classification using K means clustering and Hard Label Method 56

5.1 Short Data set: K means - Hard Label ...................................... 58
5.2 Short Data Set: K means - Fuzzy Label ................................. 59
5.3 Short Data Set: SVM ............................................................. 60
5.4 Long Data Set: K means - Hard Label ................................... 61
5.5 Long Data Set: K means - Fuzzy Label ................................. 62
5.6 Long Data Set: SVM ............................................................. 62
5.7 Accuracy Summary for all adopted methods .......................... 63
## List of Figures

2.1 Locations of sensor placements ................................................. 6

3.1 Basic Matrix Multiplication ................................................... 10
3.2 Hadamard Product ................................................................. 11
3.3 Kronecker Product ................................................................. 12
3.4 Khatri-Rao Product ................................................................. 13
3.5 Types of Tensors ................................................................. 14
3.6 Fiber tensors [5] ................................................................. 15
3.7 Slice tensors [5] ................................................................. 15
3.8 Elements of Tensor $T$ ......................................................... 16
3.9 Modes of Tensor $T$ ................................................................. 16
3.10 Inner product of two tensors .................................................. 17
3.11 Product of a tensor and a matrix ........................................... 18
3.12 Product of a tensor and a vector ........................................... 19
3.15 BTD Decomposition [14] ......................................................... 21
3.16 Recorded signal and respective spectrogram ................................. 36
3.17 Example of STFT with window length $L = 20$ ............................. 37
3.18 Example of STFT with window length $L = 10$ ............................. 37
3.19 Supervised Learning .............................................................. 40
3.20 K-means Clustering .............................................................. 42
3.21 K-means Clustering - Iteration 1,5,10 ....................................... 44
3.22 Choosing a line, plane or hyper-plane ..................................... 45
3.23 Determining Margins ............................................................ 46
3.24 Classification with Outliers .................................................... 47

4.1 Data Arrangement ............................................................... 50
4.2 Decomposition of a tensor into factor matrices using CPD ............ 51
4.3 Error plot using rankest for decomposition of tensors .................... 52
Chapter 1

Introduction

1.1 Motivation

Parkinsons Disease (PD) is a progressive disorder which occurs in the nervous system. Close to one million people in America suffer with Parkinson’s disease; of which 60,000 people are diagnosed on a yearly basis. There are several cases which go undetected besides this appalling number.

This not only affects the patients in its own way, but also ends up causing severe depression, mental trauma, alternation in mood, behavior and thoughts along with disturbed sleeping patterns. The symptoms for Parkinsons Disease can be classified into 2 broad types; namely motor symptoms and non-motor symptoms. Motor symptoms are mainly of four types: Tremor, rigidity, postural instability and slowness of movement.

Tremor is one of the most common symptoms which can be spotted in the PD patients. In some cases, the tremor is a symptom that develops as the disease progresses. Tremors generally occur during the rest posture of the patient; in other words; these are involuntary unnecessary movements of the limb or arms during rest positions. Frequency of these tremors generally lie in the region of 4-6Hz.

Hypokinesia or slowness of movement is one of the characteristic features of the Parkinsons Disease which is basically defined as the difficulty in carrying out a particular motion or movement. The patients often complain of the inability of doing daily tasks like writing, getting dressed, walking, etc.
With the information provided by the symptoms of the disease, a method is being devised to perform the assessment of Parkinson’s disease with the help of machine learning techniques. With the advancements in technology, one can make use of sensors which contain an accelerometer as well as gyroscope to monitor the movements of any action during any activity performed by an individual.

One of the primary methods used to control the severity of the problems caused by this disease is the medical drug called Levodopa.[1] The reason for the cause of these involuntary movements is the lack of production of a chemical in the body called dopamine. Dopamine is released by the neurons in the body and has the fundamental task of sending signals to the other nerve cells. It is responsible for the motor control in the human body and the release of other hormones throughout the system. The consumption of this drug helps in the reduction of the PD symptoms and also regulate the ease of activity performance, but may also cause involuntary movements called dyskinesias.

1.2 Objectives

With the advancements in sensors used for monitoring health, the interest in this field has grown to lead researchers analyze health of patients for various causes. This led to the interest in developing an algorithm which can assess the medical states of Parkinson’s disease. Previous works include the extensive analysis on the walking or non-walking movements of PD patients [11, 17, 19, 21].

Other studies [9] has spoken about the medical state classification based on mounted sensors in different parts of the body. Furthermore, none of the above listed papers are patient specific. They do not take into account the variations for every patient. As we know, PD is a disorder whose severity differs from patient to patient. Paper [20] uses unconstrained activities as the method to assess and analyze the PD patients and the concept of patient specific approach was adopted in [10]. Hence, the algorithm developed here helps overcome the generalization for patients and deals with every patient separately as individual subjects; the difference being the signal fusion concept and the methodology of tensor decomposition.
Following listed are some of the aspects which would be applied in this approach towards assessing Parkinson’s Disease patients.

1. **Sensor feedback in terms of accelerometer and gyroscope readings**
   Signals plotted against time do not provide too much information. The alternate solution would be to extract more information from a time-signal is to perform the Fourier transform of the signal and look at the time-frequency response for every patient performing various tasks.

2. **To determine and analyze specific patterns which differentiate state of patient**
   From the time-frequency response of the sensor signals, features can be extracted that relate to the symptoms of Parkinson’s disease; which help better understand the signal response for each patient individually.

3. **Perform machine learning algorithms**
   With the help of features extracted from each patient’s sensor readings, using the knowledge of Supervised and Unsupervised algorithms helps us understand and predict the accuracy of assessment of each patient suffering from Parkinson’s Disease.

1.3 **Contribution**

In this thesis, some of the methods used to complete the aforementioned objectives were as follows:

1. Use of Tensor Mathematics to keep related data together for each patient.

2. Development of an algorithm to automatically create tensors for each patient with the help of the signals produced by the sensors patients were asked to wear.

3. Development of machine learning algorithms which would help to better assess each patient for the severity of Parkinson’s disease.
1.4 Thesis Organization

The thesis is organized into six chapters.

**Chapter 1 (Introduction)** briefly explains the motivation behind this thesis work. In addition, this chapter also discusses the objectives and provides a glimpse into the study methods used in the thesis.

**Chapter 2 (Experimental Setup and Background)** talks about the experimental setup which is carried out in the clinics. This chapter is also devoted to talking about the various details of the data set which is being generated for every patient.

**Chapter 3 (Methodology)** talks about 3 broad areas which are necessary to understand the working of the project. Part one deals with the basic terminologies related to Tensors which would be necessary for the organization of data in a systematic manner. It goes briefly into the types of tensors followed by the various tensor decomposition techniques as this is necessary for the programming logic applied in feature extraction. Next Topic explained deals with the concept of Short Time Fourier Transform. Final part of this chapter deals with the basic machine learning concepts needed to perform the algorithm for determining the results related to the Parkinson’s disease assessment.

**Chapter 4 (Proposed Methods)** gives a descriptive overview on what the various techniques have been adopted for this thesis. It also gives a brief idea of the various features which are being used to determine the condition of each patient when they are asked to perform a task

**Chapter 5 (Results and Analysis)** shows the various results obtained by the algorithm and analyzes the places where improvements can be made.

**Chapter 6 (Conclusions and Future Works)** gives the reader a brief idea as to what next steps could be taken to improve the results produced in this thesis.
Chapter 2

Experimental Setup and Background

2.1 Test Setup

Fifteen PD patients with motor fluctuations participated in this experiment [16]. The patients were between the ages of 42 and 77 years; comprised of 9 males and 6 females who have had a PD history of 3.5-17 years. They also had a 13-60 Unified Parkinson’s Disease Rating Scale (UPDRS) in the OFF state. UPDRS can be defined as a scale which determines the level or extent of severity of the PD symptoms. The patients were asked to wear five KinetiSense motion sensors (Great Lakes NeuroTechnologies Inc., Cleveland, OH) mounted on left and right wrist, trunk (upper back) and left and right ankle. The motion sensors units contain an accelerometer, a gyroscope, and a magnetometer 3-d sensors.

The patients are expected to hold onto their PD medication dose before the experiment was conducted. But for certain patients, the PD effect was so advanced that they were asked to perform the tasks assigned after a hours from the consumption of the medication. The experiment was divided into 4 rounds of tasks. First round was before taking a dose of levodopa (50-350mg) [4] and the other three rounds were scheduled after one hour for a period of 3 hours and one hour gap between them. In each round, eight activities were performed which were ambulation, dressing, drinking from a cup, arm resting while setting, unpacking groceries, cutting food, and hair brushing using left and right arms. Every task was about 30 seconds and were video-taped. Each round was classified as OFF or ON state based on clinical evaluation performed at the time of each round.
The sensors are used to track the movements of the patients while they are asked to perform a particular task. The accelerometers and the gyroscopes in the sensor record every motion the patient makes while performing a series of tasks. The signal recorded is then filtered to removed the noise captured and carefully assessed to extract features related to the disease.

Some of the tasks patients are asked to perform are

1. Ambulation or Walking
2. Cutting
3. Dressing
4. Drinking water
5. Grocery sorting

A person suffering from Parkinson’s disease can be segregated into either a left sided or a right sided more affected patient. This information is provided by clinicians after carefully watching them perform the tasks assigned to them. Accordingly, the wrist and ankle
sensors are chosen for each patient. If the patient is left sided, data is extracted from the
left wrist and left arm. Since every patient varies from one another, it is very important that
the ”Most Affected Side” is determined properly.

As we know, the low dopamine production in the patient’s body is the reason for the rigidity
and other symptoms for this disease. In order to counter this effect, they are given medica-
tion in the form of Levodopa. It is a drug which brings the involuntary movements under
control while performing day-to-day tasks assigned to them.

The data recordings are then classified into 4 broad categories.

1. **R0**: Activity recorded when patient is given no medication

2. **R1**: Activity recorded 1 hour after medication has been consumed

3. **R2**: Activity recorded 2 hours after medication has been consumed

4. **R3**: Activity recorded 3 hours after medication has been consumed

### 2.2 Data sets

In order to assess the issue in the best way possible, the clinicians provided sensor record-
ings for two sets of patients.

The first data set involves sensor recordings of patients on a short-interval basis. The pa-
tients were asked to perform activities which would span 20-30 seconds. As the patients
are performing the tasks, the sensor uses the accelerometer and gyroscopes to record the
movements of the patients for every task performed. Data is extracted into a spreadsheet
for further processing.

The second data set comprises of another set of patients for whom various work stations
are set up by the clinicians where the patient is asked to perform various activities over a
longer period of time. They are asked to do day-to-day activities and are recorded over a
2 hour period. The sensors record all the information and the data is exported into another
spreadsheet which is used for further assessment.
Irrespective of the data set, a particular method needs to be followed to attain the accuracy of the assessment process. This involves the segmentation of the data set into 2 classes; namely the train data set and the test data set.

The clinicians decide the type of activities which can be considered for training and testing depending on the complexity of the task which needs to be performed. Moreover, the tasks which were selected for training were the tasks which have been carried out by the clinicians in the hospitals to assess the patient from the PD symptoms. Once the tasks have been divided into training and testing activities, the algorithm devised will perform assessment on each patient. With the help of the train data points, a model can be developed which is then tested with unseen data points to determine the accuracy and it’s ability to learn and classify new data points.

This approach will help the clinicians better understand the severity of Parkinson’s disease symptoms with the help of an algorithmic viewpoint. This can help them save time and carry out an automated process which will help work towards the assessment of the patients condition.
Chapter 3

Methodology

3.1 Matrix and Tensor Decomposition

3.1.1 Fundamentals of Matrices

A matrix is defined as a two dimensional structure which helps us keep data related together which vary in two aspects. This can be as simple as a row-column distribution of students and their test scores. The need for matrices can be explained by the ability to understand the relationship between 2 features.

This section talks about the various types of Matrix Multiplication techniques which would be useful for the multiplication process for matrices as well as higher order matrices which are called tensors.

The use of matrices brings clarity to some of the following points:

1. Data represented in the form of matrices gives us practical understanding about relationships between two features, variables or entities.

2. The manner in which the data is organized can help make easier real-life calculations.

3. Matrix multiplications could be used in simple calculations which involve day-to-day purchasing or may also involve complex high-level calculations which can be simplified using matrix multiplication.
Matrix Multiplication

Let us assume we have 2 matrices which are of the sizes (IxJ) and (KxJ) respectively. The product of the two matrices is shown below.

\[
A = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1K} \\
a_{21} & a_{22} & \cdots & a_{2K} \\
\vdots & \vdots & \ddots & \vdots \\
a_{I1} & a_{I2} & \cdots & a_{IK}
\end{bmatrix}_{I \times K}
\]

\[
B = \begin{bmatrix}
b_{11} & b_{12} & \cdots & b_{1J} \\
b_{21} & b_{22} & \cdots & b_{2J} \\
\vdots & \vdots & \ddots & \vdots \\
b_{K1} & b_{K2} & \cdots & b_{KJ}
\end{bmatrix}_{K \times J}
\]

Multiplication of the above shown matrices is given by

\[
A \times B = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1K} \\
a_{i1} & a_{i2} & \cdots & a_{ik} \\
a_{I1} & a_{I2} & \cdots & a_{IK}
\end{bmatrix}
\begin{bmatrix}
b_{11} & b_{12} & \cdots & b_{1J} \\
b_{21} & b_{22} & \cdots & b_{2J} \\
\vdots & \vdots & \ddots & \vdots \\
b_{K1} & b_{K2} & \cdots & b_{KJ}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots
\end{bmatrix}
\begin{bmatrix}
\langle A_{i*}, B_{i*} \rangle
\end{bmatrix}
\]

Figure 3.1: Basic Matrix Multiplication
Hadamard Product

It involves the multiplication of 2 matrices with the same size (IxJ). Following figure shows the Hadamard Product of the two matrices.

\[
A = \begin{bmatrix}
  a_{11} & a_{12} & \cdots & a_{1J} \\
  a_{21} & a_{22} & \cdots & a_{2J} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{I1} & a_{I2} & \cdots & a_{IJ}
\end{bmatrix}_{I \times J}, \quad
B = \begin{bmatrix}
  b_{11} & b_{12} & \cdots & b_{1J} \\
  b_{21} & b_{22} & \cdots & b_{2J} \\
  \vdots & \vdots & \ddots & \vdots \\
  b_{I1} & b_{I2} & \cdots & b_{IJ}
\end{bmatrix}_{I \times J}
\]

The Hadamard product is given by

\[
A \ast B = \begin{bmatrix}
  a_{11} & a_{12} & \cdots & a_{1J} \\
  a_{21} & a_{22} & \cdots & a_{2J} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{I1} & a_{I2} & \cdots & a_{IJ}
\end{bmatrix}_{I \times J} \begin{bmatrix}
  b_{11} & b_{12} & \cdots & b_{1J} \\
  b_{21} & b_{22} & \cdots & b_{2J} \\
  \vdots & \vdots & \ddots & \vdots \\
  b_{I1} & b_{I2} & \cdots & b_{IJ}
\end{bmatrix}_{I \times J}
\]

\[
= \begin{bmatrix}
  a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1J}b_{1J} \\
  a_{21}b_{21} & a_{22}b_{22} & \cdots & a_{2J}b_{2J} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{I1}b_{I1} & a_{I2}b_{I2} & \cdots & a_{IJ}b_{IJ}
\end{bmatrix}_{I \times J}
\]

Figure 3.2: Hadamard Product
Kronecker Product

Following figure shows the Kronecker Product of the two matrices

\[
A = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1J} \\
a_{21} & a_{22} & \cdots & a_{2J} \\
\vdots & \vdots & \ddots & \vdots \\
a_{I1} & a_{I2} & \cdots & a_{IJ}
\end{bmatrix}_{I \times J}
\]

\[
B = \begin{bmatrix}
b_{11} & b_{12} & \cdots & b_{1L} \\
b_{21} & b_{22} & \cdots & b_{2L} \\
\vdots & \vdots & \ddots & \vdots \\
b_{K1} & b_{K2} & \cdots & b_{KL}
\end{bmatrix}_{K \times L}
\]

The Kronecker product of the above two matrices is given by

\[
A \otimes B = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1J} \\
a_{21} & a_{22} & \cdots & a_{2J} \\
\vdots & \vdots & \ddots & \vdots \\
a_{I1} & a_{I2} & \cdots & a_{IJ}\end{bmatrix}
\begin{bmatrix}
b_{11} & b_{12} & \cdots & b_{1L} \\
b_{21} & b_{22} & \cdots & b_{2L} \\
\vdots & \vdots & \ddots & \vdots \\
b_{K1} & b_{K2} & \cdots & b_{KL}\end{bmatrix}
\]

\[
= \begin{bmatrix}
a_{11} & \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1L} \\
b_{21} & b_{22} & \cdots & b_{2L} \\
\vdots & \vdots & \ddots & \vdots \\
b_{K1} & b_{K2} & \cdots & b_{KL}\end{bmatrix} & \cdots & a_{1J} & \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1L} \\
b_{21} & b_{22} & \cdots & b_{2L} \\
\vdots & \vdots & \ddots & \vdots \\
b_{K1} & b_{K2} & \cdots & b_{KL}\end{bmatrix}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
a_{I1} & \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1L} \\
b_{21} & b_{22} & \cdots & b_{2L} \\
\vdots & \vdots & \ddots & \vdots \\
b_{K1} & b_{K2} & \cdots & b_{KL}\end{bmatrix} & \cdots & a_{IJ} & \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1L} \\
b_{21} & b_{22} & \cdots & b_{2L} \\
\vdots & \vdots & \ddots & \vdots \\
b_{K1} & b_{K2} & \cdots & b_{KL}\end{bmatrix}
\end{bmatrix}
\]

Figure 3.3: Kronecker Product
Khatri-Rao Product

Finally, this involves the product of two matrices which have the order as $(I \times K)$ and $(J \times K)$.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1K} \\ a_{21} & a_{22} & \cdots & a_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1} & a_{I2} & \cdots & a_{IK} \end{bmatrix}_{I \times K} \quad B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1K} \\ b_{21} & b_{22} & \cdots & b_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ b_{J1} & b_{J2} & \cdots & b_{JK} \end{bmatrix}_{J \times K}$$

The Khatri – Rao product is given by

$$A \odot B = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1K} \\ a_{21} & a_{22} & \cdots & a_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1} & a_{I2} & \cdots & a_{IK} \end{bmatrix}_{I \times K} \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1K} \\ b_{21} & b_{22} & \cdots & b_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ b_{J1} & b_{J2} & \cdots & b_{JK} \end{bmatrix}_{J \times K}$$

$$= \begin{bmatrix} a_{11} \begin{pmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{J1} \end{pmatrix} & \cdots & a_{1J} \begin{pmatrix} b_{1K} \\ b_{2K} \\ \vdots \\ b_{JK} \end{pmatrix} \\ \vdots & \ddots & \vdots \\ a_{I1} \begin{pmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{J1} \end{pmatrix} & \cdots & a_{IJ} \begin{pmatrix} b_{1K} \\ b_{2K} \\ \vdots \\ b_{JK} \end{pmatrix} \end{bmatrix}$$

Figure 3.4: Khatri-Rao Product
3.1.2 Introduction to Tensors

Tensors are considered as an emerging concept which can be applied to multi-way data fusion. It can be described as a method used to represent data of higher dimensionality. A tensor can be as simple as a matrix (two-dimensional in nature) or as complex as a cube holding data (three-dimensional in nature)[6]

Tensors can be 1-dimensional, 2-dimensional, 3-dimensional or N-dimensional in size. The following figure shows 2D and 3D tensors.

\[
\text{Second Order Tensor} \ A = (a_{ij}) \in \mathbb{R}^{n_1 \times n_2}
\]

\[
\text{Third Order Tensor} \ A = (a_{ijk}) \in \mathbb{R}^{n_1 \times n_2 \times n_3}
\]

\[
\text{p}^{\text{th}} \text{ Order Tensor} \ A = (a_{i_1i_2i_3...i_p}) \in \mathbb{R}^{n_1 \times ... \times n_p}
\]

Figure 3.5: Types of Tensors

The order of a tensor is given by the number of directions that the data is defined in. In simple words, direction refers to the different variables which are used to describe the data in context. It is also referred to as ways or modes of a tensor. A Tensor can be viewed in 2 ways.

One method to view the contents of a tensor is to consider it as long vectors taken along different directions of the tensor. This is called the Fiber. The following figure shows the row fiber, column fiber and the tube fiber.
A row fiber in other words is a row vector. A column fiber is a column vector and the tube fiber can be visually understood as the vector when looking into the third dimension of a three dimension object. In simple words, a fiber tensor can be described as a one-dimensional fragment of a tensor which can be obtained by fixing all indices of the tensor except for one.

The other method is to view them as slices as shown below. Frontal, horizontal and lateral slices are the 3 unique types of Tensor Slices.
Frontal slices can be explained as the matrix which involves the x-y plane. Horizontal slices can be the x-z plane while the Lateral slices are the ones with the y-z plane. In other words, A slice tensor is a two-dimensional portion or fragment of a tensor which can be obtained by fixing all indices of the tensor except for two.

3.1.3 Tensor Unfolding

Performing computations on a tensor with N dimensions can be very time consuming and difficult. Hence, for simplifying large computations, the tensors can be converted into matrices by a concept of Tensor unfolding which is shown in the figure below.

\[
T = \begin{bmatrix}
  t_{111} & t_{121} & t_{112} & t_{122} \\
  t_{211} & t_{221} & t_{212} & t_{222} \\
  t_{111} & t_{121} & t_{112} & t_{122}
\end{bmatrix}
\]

Figure 3.8: Elements of Tensor \(T\)

Let us assume that a Tensor consists of 2 slices with elements named according to position. For e.g. In the above figure shows the position (2,1) in slice number 2 is the 212 position. The above tensor can be unfolded in three ways.

**Mode 1:**
\[
T_{(1)} = \begin{bmatrix}
  t_{111} & t_{121} & t_{112} & t_{122} \\
  t_{211} & t_{221} & t_{212} & t_{222}
\end{bmatrix}
\]

**Mode 2:**
\[
T_{(2)} = \begin{bmatrix}
  t_{111} & t_{211} & t_{112} & t_{212} \\
  t_{121} & t_{221} & t_{122} & t_{222}
\end{bmatrix}
\]

**Mode 3:**
\[
T_{(3)} = \begin{bmatrix}
  t_{111} & t_{211} & t_{112} & t_{221} \\
  t_{112} & t_{212} & t_{121} & t_{222}
\end{bmatrix}
\]

Figure 3.9: Modes of Tensor \(T\)
Mode 1, the elements of slice 1 are following by elements of slice 2 in the matrix. In Mode 2, transpose of slice 1 is followed by transpose of slice 2. In Mode 3, slice one elements are placed in the first row of the unfolded matrix while the elements of slice 2 are placed in row 2 of the unfolded matrix.

### 3.1.4 Tensor Multiplication

Tensors can be treated as a matrix and be multiplied with another tensor, a matrix or a vector.

**Inner product of Tensors of the same size**

The product of two equal sized tensors is the product of the elements at same locations in their respective tensors. This can be explained with an example.

\[ \text{Let us consider} \]
\[
A = \begin{bmatrix} a_1 & a_2 & a_5 & a_6 \\ a_3 & a_4 & a_7 & a_8 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} b_1 & b_2 & b_5 & b_6 \\ b_3 & b_4 & b_7 & b_8 \end{bmatrix}
\]

*then the inner product of both the tensors can be defined as follows*

\[
< A, B > = a_1 b_1 + a_2 b_2 + a_3 b_3 + a_4 b_4 + a_5 b_5 + a_6 b_6 + a_7 b_7 + a_8 b_8
\]

*In general, it can be given by the formula below*

\[
< A, B > = \sum_{i=1}^{N} a_i b_i
\]

*where N is the equal number of elements in both tensors*

Figure 3.10: Inner product of two tensors
Product of a tensor and a matrix

When we consider the product of a tensor and a matrix, we have to decide the dimension of the tensor that needs to be taken into account for multiplying with the matrix. This is decided by defining the n-mode product of a tensor with a matrix.

Let us consider

\[ T = \begin{bmatrix}
1 & 2 & 5 \\
2 & 7 & 6 \\
3 & 4 & 8 \\
\end{bmatrix} \text{ and } A = \begin{bmatrix}
a & b \\
c & d \\
\end{bmatrix} \]

We want to find the mode – 1 product of Tensor \( T \) with matrix \( A \)

Hence the mode – 1 matricization is \( T_{(1)} = \begin{bmatrix}
1 & 2 & 5 & 6 \\
3 & 4 & 7 & 8 \\
\end{bmatrix} \)

Now, the tensor product is as follows

\[ P = A \times T_{(1)} = \begin{bmatrix}
a & b \\
c & d \\
\end{bmatrix} \begin{bmatrix}
1 & 2 & 5 & 6 \\
3 & 4 & 7 & 8 \\
\end{bmatrix} \]

\[ = \begin{bmatrix}
a + 3b & 2a + 4b & 5a + 7b & 6a + 8b \\
c + 3d & 2c + 4d & 5c + 7d & 6c + 8d \\
\end{bmatrix} \]

Therefore, \( P = \begin{bmatrix}
a + 3b & 2a + 4b \\
c + 3d & 2c + 4d \\
5a + 7b & 6a + 8b \\
5c + 7d & 6c + 8d \\
\end{bmatrix} \)

Figure 3.11: Product of a tensor and a matrix
Product of a tensor and a vector

When we consider the product of a tensor and a vector, we have to decide the dimension of the tensor that needs to be taken into account for multiplying with the vector. This is decided by defining the n-mode product of a tensor with a vector. It is shown in the figure below as the variable 'P'.

Let us consider

the tensor, \( T = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \) and vector \( v = \begin{pmatrix} a \\ b \end{pmatrix} \)

we want to find the mode – 1 product of \( T \) with vector \( v \).

Hence, mode – 1 matricization is \( T_{(1)} = \begin{pmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{pmatrix} \)

Now the tensor product is as follows

\[
P = T_{(1)} \otimes v = \begin{pmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}
\]

\[
= \begin{pmatrix} a & 2a & 5a & 6a \\ b & 2b & 5b & 6b \\ 3a & 4a & 7a & 8a \\ 3b & 4b & 7b & 8b \end{pmatrix}
\]

Therefore, \( P = \begin{pmatrix} a & 2a & 5a & 6a \\ b & 2b & 5b & 6b \\ 3a & 4a & 7a & 8a \\ 3b & 4b & 7b & 8b \end{pmatrix} \)

Figure 3.12: Product of a tensor and a vector
3.1.5 Tensor Decomposition

Tensors decomposition is defined as the process of breaking down a large tensor with data into smaller factor matrices such that the summation of the decomposition leads to the approximation of the original tensor.

Tensor decomposition can be broadly classified into 3 categories. They are as follows

1. Canonical Polyadic Decomposition
2. Low multi-linear rank approximation
3. Block term decomposition

Canonical Polyadic Decomposition (CPD)

It can be used to approximate a tensor with the help of summation of $R$ Rank 1 tensors as shown in the figure. Each component is called a $U$ matrix which is represented by 3 factor matrices; $a,b,c$; whose product yields a portion of the original Tensor.[6]

Figure 3.13: CP Decomposition [14]
**Low multilinear rank approximation (LMLRA)**

LMLRA is another method to approximate the original tensor using 3 factor matrices along with a single core tensor. It can be pictorially represented as shown below.

![Figure 3.14: LMLRA Decomposition [14]](image)

**Block term decomposition (BTD)**

This decomposition breaks the original tensors into $R$ components such that each component has 3 factor matrices along with a core tensor each. Each of these core tensors could have different dimensions and totally depends on the original tensor being decomposed.

![Figure 3.15: BTD Decomposition [14]](image)
3.2 Canonical Polyadic Decomposition

Following is an example of how a rank 1 tensor can be decomposed and re-computed using the Canonical Polyadic Decomposition technique.

Let us consider a tensor, 
\[ T = \begin{bmatrix} 2 & 4 \\ 6 & 12 \end{bmatrix} \]

Following are the formulae required to carry out the

Canonical Polyadic Decomposition

\[
\begin{align*}
    a &\leftarrow \tau_1(c \odot b)(c^T c * b^T b)^{-1} \\
    b &\leftarrow \tau_2(c \odot a)(c^T c * a^T a)^{-1} \\
    c &\leftarrow \tau_3(b \odot a)(b^T b * a^T a)^{-1}
\end{align*}
\]

Step 1: To use the first formula, let us assume values for vector \( b \) and \( c \)

Let \( b = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \) and \( c = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \)

\[
\begin{align*}
    a_1 &= \begin{bmatrix} 1 & 2 & 2 & 4 \\ 3 & 6 & 6 & 12 \end{bmatrix} \times \begin{bmatrix} 1 \\ -1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 1 \\ -1 \end{bmatrix}^{-1} \\
    a_1 &= \begin{bmatrix} 1 & 2 & 2 & 4 \\ 3 & 6 & 6 & 12 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 1 \\ -1 \end{bmatrix}^{-1} \\
    a_1 &= \begin{bmatrix} 1 \\ 0 \\ 1 \\ -2 \end{bmatrix} \times \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix}^{-1} = \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix} \times \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \\ 3 \\ -2 \end{bmatrix}
\end{align*}
\]

Hence, \( a_1 = \begin{bmatrix} -1 \\ 2 \\ 3 \\ -2 \end{bmatrix} \)
Step 2: Apply the second formula, using the value for \( a \) as what we got above and the initially assumed \( c \)

Let \( a = \begin{bmatrix} -\frac{1}{2} \\ \frac{3}{2} \\ -\frac{3}{2} \end{bmatrix} \) and \( c = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \)

\[
b_1 = \begin{bmatrix} 1 & \frac{3}{2} & 6 \\ 2 & 6 & 4 \\ 6 & 12 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} \odot \begin{bmatrix} -\frac{1}{2} \\ \frac{3}{2} \\ \frac{3}{2} \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} \times \begin{bmatrix} -\frac{1}{2} & -\frac{3}{2} \\ -\frac{3}{2} & -\frac{3}{2} \end{bmatrix}^{-1}
\]

\[
b_1 = \begin{bmatrix} 1 & \frac{3}{2} & 6 \\ 2 & 6 & 4 \\ 6 & 12 \end{bmatrix} \begin{bmatrix} 1/2 \\ -3/2 \\ 0 \\ 0 \end{bmatrix}^{(5/2)^{-1}} = \begin{bmatrix} -5 \\ -10 \end{bmatrix} \times \begin{bmatrix} 5/2 \\ -5/2 \end{bmatrix} = \begin{bmatrix} -2 \\ -4 \end{bmatrix}
\]

Hence, \( b_1 = \begin{bmatrix} -2 \\ -4 \end{bmatrix} \)

Step 3: Apply the third formula, using the value for \( a \) and \( b \) from Step 2

Let \( a = \begin{bmatrix} -\frac{1}{2} \\ \frac{3}{2} \\ -\frac{3}{2} \end{bmatrix} \) and \( b = \begin{bmatrix} -2 \\ -4 \end{bmatrix} \)

\[
c_1 = \begin{bmatrix} 1 & \frac{3}{2} & 6 \\ 2 & 6 & 4 \\ 6 & 12 \end{bmatrix} \times \begin{bmatrix} -2 \\ -4 \end{bmatrix} \odot \begin{bmatrix} -\frac{1}{2} \\ \frac{3}{2} \\ \frac{3}{2} \end{bmatrix} \times \begin{bmatrix} -2 \\ -4 \\ -2 \end{bmatrix} \times \begin{bmatrix} -\frac{1}{2} & -\frac{3}{2} \\ -\frac{3}{2} & -\frac{3}{2} \end{bmatrix}^{-1}
\]

\[
c_1 = \begin{bmatrix} 1 & \frac{3}{2} & 6 \\ 2 & 6 & 4 \\ 6 & 12 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 2 \\ 6 \end{bmatrix} \begin{bmatrix} 20 \times \frac{5}{2} \end{bmatrix}^{-1} = \begin{bmatrix} 50 \\ 100 \end{bmatrix} \times \begin{bmatrix} 1/50 \\ 1/50 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}
\]

Hence, \( c_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \)
Step 4: To confirm that the tensor is a rank - 1 tensor, we will use \( a = a_1, b = b_1 \) and \( c = c_1 \)

Let \( b = b_1 = \begin{bmatrix} -2 \\ -4 \end{bmatrix} \) and \( c = c_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \)

\[
a_2 = \begin{bmatrix} 1 & 2 & 2 \\ 3 & 6 & 6 \\ 12 \end{bmatrix} \times \left( \begin{bmatrix} 1 \\ 2 \end{bmatrix} \odot \begin{bmatrix} -2 \\ -4 \end{bmatrix} \right) \times \begin{bmatrix} 1 \\ 2 \\ [-2 & -4] \end{bmatrix} \begin{bmatrix} -2 \\ -4 \end{bmatrix} \right)^{-1}
\]

\[
a_2 = \begin{bmatrix} 1 & 2 & 2 \\ 3 & 6 & 6 \\ 12 \end{bmatrix} \begin{bmatrix} -2 \\ -4 \\ -8 \end{bmatrix} (5 \times 20)^{-1} = \begin{bmatrix} -50 \\ -150 \end{bmatrix} \times \left( \begin{bmatrix} -1 \\ 100 \end{bmatrix} \right) = \begin{bmatrix} -1/2 \\ -3/2 \end{bmatrix}
\]

Hence, \( a_2 = a_1 = \begin{bmatrix} -1/2 \\ -3/2 \end{bmatrix} \)

Step 5: To find \( b_2 \)

Let \( a = a_1 = \begin{bmatrix} -1/2 \\ -3/2 \end{bmatrix} \) and \( c = c_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \)

\[
b_2 = \begin{bmatrix} 1 & 3 & 2 & 6 \\ 2 & 6 & 4 & 12 \end{bmatrix} \times \left( \begin{bmatrix} 1 \\ 2 \end{bmatrix} \odot \begin{bmatrix} -1/2 \\ -3/2 \end{bmatrix} \right) \times \begin{bmatrix} 1 \\ 2 \\ [-1/2 & -3/2] \end{bmatrix} \begin{bmatrix} -1/2 \\ -3/2 \end{bmatrix} \right)^{-1}
\]

\[
b_2 = \begin{bmatrix} 1 & 3 & 2 & 6 \\ 2 & 6 & 4 & 12 \end{bmatrix} \begin{bmatrix} -1/2 \\ -3/2 \\ -1 \\ -3 \end{bmatrix} \begin{bmatrix} 5 \times 5 \end{bmatrix}^{-1} = \begin{bmatrix} -25 \\ -50 \end{bmatrix} \times \left( \begin{bmatrix} 2 \\ 25 \end{bmatrix} \right) = \begin{bmatrix} -2 \\ -4 \end{bmatrix}
\]

Hence, \( b_2 = b_1 = \begin{bmatrix} -2 \\ -4 \end{bmatrix} \)
Step 6: To find $c_2$

Let $b = b_1 = \begin{bmatrix} -2 \\ -4 \end{bmatrix}$ and $c = c_1 = \begin{bmatrix} -1/2 \\ -3/2 \end{bmatrix}$

$$b_2 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 6 \\ 4 \\ 12 \end{bmatrix} \times \begin{bmatrix} -2 \\ -4 \end{bmatrix} \odot \begin{bmatrix} -1/2 \\ -3/2 \end{bmatrix} \times \begin{bmatrix} -2 \\ -4 \end{bmatrix} \begin{bmatrix} -1/2 \\ -3/2 \end{bmatrix}^{-1}$$

$$b_2 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 6 \\ 4 \\ 12 \end{bmatrix} \left( \begin{bmatrix} 1 \\ 3 \\ 2 \\ 6 \end{bmatrix} \begin{bmatrix} 20 \times \frac{5}{2} \end{bmatrix} \right)^{-1} = \begin{bmatrix} 50 \\ 100 \end{bmatrix} \times \left( \begin{bmatrix} 1 \\ 50 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Hence, $c = c_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

Since $a_1 = a_2$, $b_1 = b_2$ and $c_1 = c_2$, the iterations to find $a$, $b$ and $c$ end here as convergence takes place. Thus we can write the tensor as follows

$$T = \begin{bmatrix} 2 & 4 \\ 1 & 6 \\ 2 & 12 \end{bmatrix} = \begin{bmatrix} -1/2 \\ -3/2 \end{bmatrix} \odot \begin{bmatrix} -2 \\ -4 \end{bmatrix} \odot \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Check: In order to determine if the vector attained are correct, we multiply them in the following manner $\Rightarrow$

$$T' \equiv T = \begin{bmatrix} 1 \\ -2 \\ -4 \end{bmatrix} \odot \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 6 \end{bmatrix} \odot \begin{bmatrix} 2 \\ 4 \\ 12 \end{bmatrix}$$

Hence $T' = T = \begin{bmatrix} 2 & 4 \\ 1 & 6 \\ 2 & 12 \end{bmatrix}$
Canonical Polyadic Decomposition (For any generalized tensor)

This figure shows a tensor $T$ given by $T_{ijk}$

$k$ (vector $c$)  \hspace{3cm}  j$ (vector $b$)

\hspace{3cm}  i$ (vector $a$)

Formulae used here are as follows:

\[
\begin{align*}
    a &\leftarrow \tau_1(c \odot b)(c^Tc + b^Tb)^{-1} \\
    b &\leftarrow \tau_2(c \odot a)(c^Tc + a^Ta)^{-1} \\
    c &\leftarrow \tau_3(b \odot a)(b^Tb + a^Ta)^{-1}
\end{align*}
\]

Matricization of a tensor could be done in three ways

\[
T_1 = \begin{bmatrix}
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\end{bmatrix}_{i \times j \times k} \quad T_2 = \begin{bmatrix}
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\end{bmatrix}_{j \times i \times k} \quad T_3 = \begin{bmatrix}
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\end{bmatrix}_{k \times i \times j}
\]

Three formulae which needs to be used are as follows:

\[
\begin{align*}
    a &\leftarrow \tau_1(c \odot b)(c^Tc + b^Tb)^{-1} \\
    b &\leftarrow \tau_2(c \odot a)(c^Tc + a^Ta)^{-1} \\
    c &\leftarrow \tau_3(b \odot a)(b^Tb + a^Ta)^{-1}
\end{align*}
\]

Repeat the update process of $a$, $b$ and $c$ until they converge
3.3 Low Rank Multi-linear Rank Approximation

Following example explains the tensor decomposition technique of LMLRA with the help of an example. It also shows how the reconstruction of the original tensor from the decomposed components is carried out.

The goal of this decomposition technique is to carry out tensor decomposition in a manner such that we can express the original tensor in the form of a core tensor and the factor matrices.

**Step 1:** Consider a Tensor $T$ as shown below. We first determine the various modes of this tensor $T$

$$T = \begin{bmatrix}
1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 \\
1 & 0 & 1 & 1
\end{bmatrix}_{2 \times 2 \times 2}$$

$$T_{(1)} = \begin{bmatrix}
1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1
\end{bmatrix} \quad T_{(2)} = \begin{bmatrix}
1 & 1 & 0 & 1 \\
0 & 1 & 0 & 1
\end{bmatrix} \quad T_{(3)} = \begin{bmatrix}
1 & 1 & 1 & 0 \\
0 & 1 & 0 & 1
\end{bmatrix}$$
Step 2: We perform singular value decomposition (SVD) on the three modes of tensors $T$. This gives us $U, S$ and $V$ matrices.

\[
\begin{array}{c}
\text{SVD}(T_{(1)}) \\
U_1 S_1 V_1 \\
\text{SVD}(T_{(2)}) \\
U_2 S_2 V_2 \\
\text{SVD}(T_{(3)}) \\
U_3 S_3 V_3
\end{array}
\]

Step 3: Core tensor $G$ can be computed using the following formulae

\[
G = T x_1 (U_1)^T x_2 (U_2)^T x_3 (U_3)^T \\
= \{(U_1)^T T_1\} x_2 (U_2)^T x_3 (U_3)^T \\
= \{(U_2)^T P_2\} x_3 (U_3)^T \\
= \{(U_3)^T Q_3\}
\]

where $P = (U_1)^T T_{(1)}$

$Q = (U_2)^T P_{(2)}$

$Q_3 = \text{mode 3 of tensor } Q$
Step 4: Find all elements in the above given formula

Note: $x_1$ corresponds to the mode 1 form of the tensor

$$T = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}_{2 \times 2 \times 2}$$

$$U_1 = \begin{bmatrix} -0.52 & -0.85 \\ -0.85 & 0.52 \end{bmatrix}; \quad U_1^T = \begin{bmatrix} -0.52 & -0.85 \\ -0.85 & 0.52 \end{bmatrix}$$

$$U_2 = \begin{bmatrix} -0.78 & -0.61 \\ -0.61 & 0.78 \end{bmatrix}; \quad U_2^T = \begin{bmatrix} -0.78 & -0.61 \\ -0.61 & 0.78 \end{bmatrix}$$

$$U_3 = \begin{bmatrix} -0.85 & -0.52 \\ -0.52 & 0.85 \end{bmatrix}; \quad U_3^T = \begin{bmatrix} -0.85 & -0.52 \\ -0.52 & 0.85 \end{bmatrix}$$

Step 5: Find $P_1 = T x_1 U_1^T \Rightarrow (U_1)^T T (x_1)$

$$P_1 = \begin{bmatrix} -0.52 & -0.85 \\ 0.85 & 0.52 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

$$P_1 = \begin{bmatrix} -1.37 & -0.52 & -0.85 & -0.85 \\ -0.32 & -0.85 & 0.52 & 0.52 \end{bmatrix}$$

$$P_{\text{original tensor}} = \begin{bmatrix} -0.85 & -0.85 \\ 0.52 & 0.52 \end{bmatrix}$$
Step 6: Find $Q_3$ using $P_2$

$$P_{\text{original tensor}} = \begin{bmatrix} -1.37 & -0.52 & -0.85 & -0.85 \\ -0.32 & -0.85 & 0.52 & 0.52 \end{bmatrix}$$

$$P_2 = \begin{bmatrix} -1.37 & -0.32 & -0.85 & 0.25 \\ -0.52 & -0.85 & -0.85 & 0.52 \end{bmatrix}$$

$$Q_{(2)} = \begin{bmatrix} -0.78 & -0.61 & -1.37 & -0.32 \\ -0.61 & 0.78 & -0.52 & -0.85 \end{bmatrix} \begin{bmatrix} -0.85 & -0.85 \\ 0.52 & 0.52 \end{bmatrix}$$

$$Q_{(2)} = \begin{bmatrix} 1.4 & 0.77 & 1.19 & -0.73 \\ 0.43 & -0.47 & -0.14 & 0.09 \end{bmatrix}$$

$$Q_{\text{original tensor}} = \begin{bmatrix} 1.4 & 0.43 & 1.19 & -0.73 \\ 0.77 & -0.47 & -0.73 & 0.09 \end{bmatrix}$$

$$Q_{(3)} = \begin{bmatrix} 1.4 & 0.77 & 0.43 & -0.47 \\ 1.19 & -0.73 & -0.14 & 0.09 \end{bmatrix}$$

Step 7: Find $G$ using $Q_3 \Rightarrow G = [(U_3)^T \cdot Q_3]$

$$G = \begin{bmatrix} -0.85 & -0.52 \\ -0.52 & 0.85 \end{bmatrix} \begin{bmatrix} 1.4 & 0.77 & 0.43 & -0.47 \\ 1.19 & -0.73 & -0.14 & 0.09 \end{bmatrix}$$

$$G_{(3)} = \begin{bmatrix} -1.82 & -0.27 & -0.29 & 0.35 \\ 0.27 & -1.03 & -0.35 & 0.32 \end{bmatrix}$$
Step 8: Find $G$ using $G_3$

$$G_{(3)} = \begin{bmatrix} -1.82 & -0.27 & -0.29 & 0.35 \\ 0.27 & -1.03 & -0.35 & 0.32 \end{bmatrix}$$

$$G_{\text{original tensor}} = \begin{bmatrix} 0.27 & -0.35 \\ -1.03 & 0.32 \\ -1.82 & -0.29 \\ -0.27 & 0.35 \end{bmatrix}$$

Hence tensor $T$ can be broken into a core tensor and 3 factor matrices $U_1, U_2$ and $U_3$

$$T = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}_{2 \times 2 \times 2} \Rightarrow \begin{cases} G = \begin{bmatrix} -1.82 & -0.29 \\ -0.27 & 0.35 \\ -1.03 & 0.32 \end{bmatrix} \\ U_1 = \begin{bmatrix} -0.52 & -0.85 \\ -0.85 & 0.52 \end{bmatrix} \\ U_2 = \begin{bmatrix} -0.78 & -0.61 \\ -0.61 & 0.78 \end{bmatrix} \\ U_3 = \begin{bmatrix} -0.85 & -0.52 \\ -0.52 & 0.85 \end{bmatrix} \end{cases}$$
Check: Use the decomposed matrices to determine the reconstructed tensor

\[
G = \begin{bmatrix}
-1.82 & -0.29 \\
-0.27 & 0.35 \\
0.27 & -1.03 \\
-0.35 & 0.32 \\
\end{bmatrix}
\]

\[
U_1 = \begin{bmatrix}
-0.52 \\
-0.85 \\
\end{bmatrix}, \quad U_2 = \begin{bmatrix}
-0.78 \\
-0.61 \\
\end{bmatrix}, \quad U_3 = \begin{bmatrix}
-0.85 \\
-0.52 \\
\end{bmatrix}
\]

\[
T = G x_1 (U_1) x_2 (U_2) x_3 (U_3)
\]

\[
= ((U_1)^T G_1) x_2 (U_2) x_3 (U_3)
\]

\[
= ((U_2)^T A_2) x_3 (U_3)
\]

\[
= ((U_3)^T B_3)
\]

\[
A_1 = U_1 x G(1) = \begin{bmatrix}
-0.52 & -0.85 \\
-0.85 & 0.52 \\
\end{bmatrix} \begin{bmatrix}
-1.82 & -0.29 \\
-0.27 & 0.35 \\
-1.03 & 0.32 \\
\end{bmatrix}
\]

\[
A_{\text{original tensor}} = \begin{bmatrix}
1.19 & -0.14 \\
-0.31 & 0.47 \\
1.4 & 0.43 \\
\end{bmatrix}
\]
Step 9: Find $B_3$ using $A_2$

$$A_{\text{original tensor}} = \begin{bmatrix}
1.19 & -0.14 & 1.02 & -0.09 \\
0.31 & 0.47 & & \\
1.4 & 0.43 & \\
1.4 & 1.02 & -0.09 \\
\end{bmatrix}$$

$$A_2 = \begin{bmatrix}
1.19 & 1.4 & 1.02 & -0.09 \\
-0.14 & 0.43 & -0.31 & 0.47 \\
\end{bmatrix}$$

$$B_2 = U_2 \times A_{(2)}$$

$$B_2 = U_1 \times G_{(1)} = \begin{bmatrix}
-0.52 & -0.85 & -1.82 & -0.29 & 0.27 & -0.35 \\
-0.85 & 0.52 & -0.27 & 0.35 & -1.03 & 0.32 \\
\end{bmatrix}$$

$$B_2 = \begin{bmatrix}
-0.85 & 1.37 & -0.75 & -0.04 \\
-0.85 & -0.52 & -0.7 & 0.56 \\
\end{bmatrix}$$

$$B_{\text{original tensor}} = \begin{bmatrix}
-0.75 & -0.7 \\
-0.04 & 0.56 \\
-0.85 & -0.85 \\
1.37 & -0.52 \\
\end{bmatrix}$$
Step 10: Find $T$ using $B_3$

$B_{\text{original tensor}} = \begin{bmatrix}
-0.85 & -0.85 & 1.37 & -0.52 \\
-0.75 & -0.7 & -0.04 & 0.56 \\
\end{bmatrix}$

$B_3 = \begin{bmatrix}
-0.85 & -0.85 & 1.37 & -0.52 \\
-0.75 & -0.7 & -0.04 & 0.56 \\
\end{bmatrix}$

$T_3 = U_3 \times B_{(3)}$

$T_3 = U_3 \times B_{(3)} = \begin{bmatrix}
-0.85 & -0.52 \\
-0.52 & 0.85 \\
\end{bmatrix} \begin{bmatrix}
-0.85 & -0.85 & 1.37 & -0.52 \\
-0.75 & -0.7 & -0.04 & 0.56 \\
\end{bmatrix}$

$T_3 = \begin{bmatrix}
1.11 & 1.19 & -0.19 & -0.15 \\
-0.19 & 0.68 & 0.68 & 0.75 \\
\end{bmatrix}$

$T_{\text{reconstructed}} = \begin{bmatrix}
-0.19 & -0.15 \\
0.68 & 0.75 \\
1.11 & 1.19 \\
-0.19 & 0.68 \\
\end{bmatrix}$
3.4 **Short Time Fourier Transform (STFT)**

Short Time-Fourier Transform or Short-Term Fourier transform is a method to determine the time-frequency response of any signal e.g. audio, sensor, etc. The manner in which STFT is carried out is that the large signal is broken down into small segments of equal lengths and then the Fourier transform is performed over each segment individually. In other words, it is a method used to analyze the frequency component or content of a signal when it varies with time.[18, 22, 24]

The spectrogram function is a unique way to visualize the time frequency variation of a signal in a single plot. It uses different colors in a plot to indicate the values of signal power in the STFT. Darker the colors, lesser the information in that segment whereas lighter the color or hue, more the information in the segment.

This can be clearly observed in the plot below. The yellow color indicates higher values returned after the Short Time Fourier Transform is applied while the blue color indicates lesser information from the signal. Bright yellow indicates higher power levels whereas bright blue indicates lower power levels on the figure below.
As we can see, while observing a signal with multiple frequencies in the time domain, it is not easy to determine the various frequencies present in the signal. These kind of signals could be found in speech, music, seismology and even in signals generated and recorded in activities performed by the human brain.

Short Time Fourier Transform can be carried out by segmenting a signal into smaller portions. It determines the tiling of the time-frequency plane where the size of each tile is given by the time and frequency resolution of the STFT.
For example, in the figures shown in this section, if we take the size of the window as $L = 20$ samples, then we have the time axis segmented into multiples of 20 with the total length depending on the length of the signal. The frequency axis is then segmented into parts which are $\frac{2\pi}{20}$ since $L = 20$. Hence each tile in the figure shown has one dimension which is of size 20 and the other dimension as $\frac{2\pi}{20}$ such that the area of each tiles remains a constant which can be determined as the summation of all the STFT values in that region.

If the window is shortened to $L = 10$, the segmentation along the frequency axis is now $\frac{2\pi}{10}$ as $L$ has been reduced. Hence, it is easy to observe that no matter what the length of the window is assumed, the area of each tile in the spectrogram plot will be $2\pi$. 

Figure 3.17: Example of STFT with window length $L = 20$

Figure 3.18: Example of STFT with window length $L = 10$
The time resolution can be said to be \( L \); where while the frequency resolution can be described as \( \frac{2\pi}{L} \). The product of the two yields \( 2\pi \) as the area of each tile. This throws light on the uncertainty principle since there is always a trade off between the time or the frequency resolution. In order to better understand the STFT of any signal using a spectrogram, one needs to compromise on either time or frequency to analyze the signal in the Fourier domain.[25]

The next important task while performing the STFT is to determine the type of window which should be used to determine the spectrogram of the signal. Some of the different windows on offer are listed below. [15]

1. Hamming Window
2. Hanning Window
3. Blackman Window
4. Bartlett Window
5. Triangular Window

For high precision analysis of time sequence signals, the most preferred window is the Hanning or the Hamming window functions. This thesis makes use of the hanning window to carry out the Short Time Fourier Transform to determine the spectrogram of the signals being used. In this case, the STFT is performed on the signals from the sensors and then stored in the form of a tensor.
3.5 Introduction to Machine Learning

Machine Learning is an evolution of computer science which helps with the study of recognizing patterns and learning computational theory. It is a method by which can be used to train a computer to learn about something with programming it.

For example, a picture of a house lit with decorative lighting with a snowy background can easily be apprehended as the period of Christmas and New Year’s. Now developing a system, which can look at this picture and interpret it given it lacks the human brain, is machine learning. Extracting various features from an image can be done with the help of machine learning algorithms. This process of training and developing of machines to analyze the patterns and carry out predictions can be described as a machine learning system.

Some of the fields in which machine learning could provide a good fit would be as follows

1. **Manufacturing**: Maintenance and manufacturing various industrial equipment.
2. **Security**: Fraud detection using pattern recognition.
4. **Health-care**: Analyzing patterns in images and signal files.

3.5.1 Types of Machine Learning Algorithms

Machine Learning techniques can be divided into various categories of which some of the most important types are given as follows

**Supervised Learning**

Supervised Learning can be described as a type of algorithm which learns from some initial information that is provided. In other words, the input data which is fed to the system has a known label. With the help of the current input data and its respective label, one can correctly classify future unknown data.
The system is taught or trained with the help of a few examples and this training process continues till a particular training accuracy is attained. Once the model has been devised, it is ready to see data which it has not seen before. This is known as the testing phase and this tells the performance of the system to unknown data.

Unsupervised Learning

Unsupervised learning is the exact opposite of the supervised learning. The difference here is that the input data which is fed to the model has no labels. Here, the model looks at all the input data points and tries to devise a method to distinguish between the data points. The algorithm looks at the points and tries to separate them into classes on grounds of certain features. One of the most common example for this type of learning is k means clustering.

Semi - Supervised Learning

Here the data which is provided as an input has both labeled as well as unlabeled data. The model then uses the labeled data to determine the class in which the unlabeled data falls by
finding patterns which would help classify the remaining data points. Some examples of this type of machine learning includes classification and regression. Classification models are those which segregate input data points into discrete variable classes; whereas regression models are based on a continuous random variable. The regression model analyzes the data points and compares it with known points which are similar to provide a prediction or label for the new entity.

3.5.2 K-means Clustering

One of the methods used in this research is the k means algorithm [12]. It is an unsupervised machine learning algorithm which is used to classify data points using some similarity. It is also known as a clustering algorithm. It is a simple way to group similar data into clusters which are labeled according to it’s cluster members.

This is a method by which the automatic clustering of data can be carried out depending on the similarity of some sample data. K means clustering is an iterative algorithm which begins either by the random initialization of the cluster centroids or by fixing some point in the data set as the centroid. It then updates the centroid position after assigning a cluster to the current data point by calculating the mean of the distance between the data point and the previous centroid.

The best way to begin this process is to define a space in which the points would be lying in. For example, if there are 2 features considered for a particular data set, then the space to be used is a 2-D space. If the number of features were 3, then we would be using a 3-D space to visualize the clustering. Similarly, if we have N features, we would be using an N-D space which can not be visually imagined.

Initially, let us assume we have 2 features in consideration. Let us say we have 300 points. Now we can visualize the features on a simple plot on Matlab as shown below.
Figure 3.20: K-means Clustering

As we can see, there are 3 clusters which are visibly distinguishable. Not always will it be evident how many clusters we must assume [13]. In this case, it is visible 3 clusters are ideal to separate the unknown data. Hence by feeding the number of clusters, the k means algorithm randomly initializes 3 cluster centroids. These cluster centroids are fixed.

Next, the algorithm takes a point from the data set and determines which is the closest cluster it can be classified into. Once that is done, the cluster centroid is moved to the mean of the previous centroid and the first data point. This carries on until all the data points are classified into a cluster. Following images shows the exact working of the algorithm for the above shown example.
As we can see from the above figures, the cluster centroid moves depending on the points which are assigned to the cluster. As this happens, the algorithm learns the various distances at which a point could be present and be assigned to a particular cluster. Once all the training points are clustered, a new set of points called the testing points are fed to the system to test how accurately it can classify the points. Euclidean distance between the test points and the latest cluster centroids will be determined; which will then help determine which cluster the test data point falls into.[2]
3.5.3 Support Vector Machines

A Support Vector Machine (SVM) is a supervised learning algorithm which tries to determine an optimal hyper-plane to categorize new examples from the testing set.

Linear SVM

Let us assume we have a data set described the way it is shown below. Here there exists 2 distinct groups of data which can be separated by a line. However, the 3 lines A,B,C are just a few options that we have out of the numerous lines we can create between the 2 classes which can help in the class separation. The main goal in SVM would be to find the most suitable line (in 2-D), plane (in 3-D) or hyper-plane (in N-D) which would help distinguish between classes easily.

![Figure 3.22: Choosing a line, plane or hyper-plane](image)

The SVM algorithm aims to maximize the distance between the data points of either classes. This distance can be defined as the Margin.
If we look carefully at the image shown above, one can say that the margin for hyper-plane B is the maximum when compared with planes A and C. This means, larger the margin, the lower the chances of mis-classification of data points. Hence, the algorithm always chooses the hyper-plane (line in this case) to be the one which has the largest margin.

It is also very important to understand that there are several lines which can be drawn parallel to the lines shown in the figure which could also be defined as a hyper plane for this example. The algorithm looks into the various possibilities and chooses the most suitable hyper-plane for classification purpose.

The image shown below looks into the scenario where data points of 2 groups are present. Now the algorithm will consider another factor which is called the outliers. In this scenario, it will choose hyper plane B because this plane manages to maximize the margin criterion.
Figure 3.24: Classification with Outliers

Hyper plane A classifies all the points correctly but then compromises on the terms of margin. Besides, this also minimizes the space which could be used to classify a new point as a 'black diamond' as a 'blue star'. Hence, it is easy to say that the SVM classifier has the ability to ignore outliers; i.e. ignore points which are far away from the main group of points it actually belongs to.

Non-linear SVM

So far, the above examples cover the classification of points in N-D spaces with a linear hyper plane. Now if we had a data point space like the one in the image below.
There is no linear hyper plane present which will separate the data properly. The SVM algorithm has to model a different kernel to the one which is used in the linear classification model. These functions manage to transform low dimensional input data space into higher dimensionality in order to classify the points with ease.

In simple words, it performs complex data transformations in order to carry out the classification process to separate the data into individual classes. In the above example, a circle would be considered as the hyper plane which can be used to classify the data points properly.
Chapter 4

Proposed Methods

4.1 Tensors Organization

If we carefully examine the first data set, the following table shows a clear picture as to how the data is initially organized with the help of tensors. Every cell in the figure is a representation of a tensor. For example, the cell common to R0 and HairBrush (L) contains a tensor which hold the spectrogram response of the signals related to the task of ambulation. HairBrush (L) is a task where the patient is asked to comb their hair using their left hand.

Table 4.1: State based activity categorization

<table>
<thead>
<tr>
<th>Activity\State</th>
<th>R0</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambulation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arm Resting</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cutting</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dressing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drinking</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Groceries</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hair Brush (L)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hair Brush (R)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Each tensor holds data from the wrist, truck and the ankle sensors. The wrist sensor data is stored in the first slice of the tensor, the trunk data in the second slice and the ankle data in the last slice of the tensor. To better imagine this tensor, assume a Rubik’s cube containing data in every cell on the cube.
4.1 Data Arrangement

The above figure shows how data is organized in the form of tensors. For this thesis we have used 3 sensors. The goal is to organize the spectrogram content of each sensor on a individual slices of the tensor respectively. The first step carried out is the short time Fourier transform of the wrist sensor. It is then stored in the first slice of a 3 dimensional tensor. Next we determine the spectrogram of the trunk sensor and store it in the second slice of the same tensor. Lastly, the ankle sensor spectrogram is stored in the third slice. As we see, the x axis represents the variation with respect to time, y axis represents the variation with respect to frequency and z-axis represents the senors being aligned.

4.2 Tensor Decomposition

The above shown figure shows each tensor in the form of a cell. Each tensor is then decomposed using the tensorlab toolbox into its corresponding factor matrices. For example, if the size of a tensor is 513x40x3. The first index accounts for the frequency components in the signal achieved from the spectrogram. The second index defines the time vector which was attained after the spectrogram of the signal was taken. The third index indicates the number of slices in the tensors. This was defined as three due to the utilization of three sensor data.
The following figure shows us an example of how the tensor decomposition is carried out in this thesis. Let us assume we have a tensor of the size as shown in the figure (513x40x3). If this tensor is broken down, we arrive at 3 factor matrices A, B and C. The number of components into which the decomposition takes place is given to be 'R'. The decomposition takes place such that we have R groups of factor matrices each containing its respective A, B and C parts.[23]

![Diagram of tensor decomposition](image)

**Figure 4.2: Decomposition of a tensor into factor matrices using CPD**

Finally, the tensor decomposition algorithm gives us 3 factor matrices A, B and C arranged in manner such that all 'R' A factor matrices are grouped together, 'R' B factor matrices are grouped together and 'R' C factor matrices are grouped together. As seen in the figure, A₁ is a matrix representing the frequency vectors. Hence it’s size would be 513x1. Similarly, A₂ will also be 513x1. Therefore the A factor matrix will be organized such that we have its size as (513x'R'). The same will be done for factor matrices B and C. If R is assumed to be 20, then the factor matrices will have the following sizes:

1. **Size of Factor Matrix, A** = 513x20
2. **Size of Factor Matrix, B** = 40x20
3. **Size of Factor Matrix, C** = 3x20

A function by the name rankest determines the number of components the original tensor can be decomposed to without losing much data [26]. The error factor plot can be
seen below which describes the error rate with the number of decomposition performed. The algorithm breaks the tensor sequentially from 2 all the way to a value 'R=17' which satisfies the error rate fixed at 0.01.

![Error plot using rankest for decomposition of tensors](image)

**Figure 4.3: Error plot using rankest for decomposition of tensors**

### 4.3 Feature Selection and Extraction

To understand the concept of feature selection and feature extraction, let us take an example of sorting a group of cars. Now, the few things which will help decide the final outcome would come down to various features; some of which are brand, model, color, year of manufacture, transmission type, class of vehicle, etc. These features will be the stepping stones which would be used to classify the cars into separate groups.

First, the CPD is performed on each structured tensor to obtain the factored matrices $A$, $B$, and $C$. 

```plaintext
A[
B[
C[
]]
]
B, and C. The number of components in each matrix depends on the rank of the tensor (i.e., \( R \)). One feature vector is extracted from the corresponding components in each matrix. For example, for components \( r \) in each factored matrix (i.e., \( A(:,r) \), \( B(:,r) \), and \( B(:,r) \)), we extract one feature vector. As a result, for every 5 second-segment of data with rank \( R \) tensor, we extract \( R \) feature vectors. The extracted features in every feature vector are selected such that they can differentiate between dyskinesia (i.e., abnormal involuntary movements at peak dose effect) and bradykinesia (i.e., reduction in the speed of spontaneous purposeful movements) in ON and OFF medication states, respectively. The following features are extracted for each feature vector \( r \). [7]

**Mean Frequency**

Mean frequency is a feature which can be defined as the average frequency around which most of the data post the decomposition of the tensors is concentrated. The formula which depicts the use of this feature can be seen below.

\[
MF = \frac{F_s \sum_{n=1}^{N} n \cdot A(n,r)}{\sum_{n=1}^{N} A(n,r)}
\]  

(4.1)

where \( F_s \) is the sampling frequency, \( N \) is the number of samples in each FFT, and \( A(:,r) \) represents a single column vector in factor Matrix \( A \).

**Average Jerk**

This is the average time that a patient involuntarily has jerks in the wrist or other parts of the body while performing a task. This is a feature which is calculated from the factor matrix B as the derivative of a single vector present in the factor matrix B.

**Sparsity values of Factor Matrices A and B**

Sparsity value can be defined as the amount of unknown content present in a given vector. This sparsity feature takes a value of zero if the vector contains one single zero element. This feature is carried out on the factor matrices A and B since they correspond to the frequency and time decomposition of the tensor.

\[
\text{Sparsity} = \frac{\sqrt{N} - \sum_{n=1}^{N} A(n,r)}{\sqrt{N} - 1}
\]  

(4.2)
**Signal Power**

Signal Power is a feature vector which determines the amount of power in a signal. The signal power indicates the amount of energy spent by the patient in doing a particular task at a given instant of time. It gives a better meaning to the signal which was provided and also helps analyze the reason behind the power value. Signal Power has been divided into different features based on the frequency bands which were suggested by clinicians. Signal power is calculated in the 1-4Hz region and 4-6Hz region.

Since most of the frequencies which are seen in the movements cause during the activities performed by patients are not more than 6 Hz, they band of interest is in the 1-6 Hz region. Anything above or below that would be considered as disturbances during the recording process.

**Factor matrix C**

As we know, the tensor decomposition returns 3 factor matrices of which the third factor matrix has not been discussed about. As we know, this matrix relates to the sensor from which the data has been extracted. Hence, the values present in this matrix were considered as a separate feature in the feature space.
4.4 Labeling Methodology

Once the data points are classified/assigned to a particular cluster, the next step which is important in determining the correctness of the classification is carried out by the labeling. Hard labeling is one of the 2 methods which will be discussed in this section.[7]

4.4.1 Hard Label

Hard labeling is a method which counts the number of votes which has been attained post the clustering process within a single group or cluster. In this case, clusters are classified as OFF clusters, ON clusters or Common Clusters.

OFF clusters are those clusters which possess more data points which were labeled as OFF after the classification process. ON clusters are those which contain more ON labeled data points in the group of data points assigned to that cluster. A common cluster is one which has both OFF and ON data points such that it becomes difficult to determine if the cluster is an OFF cluster or ON cluster. Following table will provide more clarity on the three types of Cluster. In the Algorithm Output column, 'one' indicates the cluster is an OFF cluster, 'two' indicates an ON Cluster and 'zero' indicates a common cluster.

In the table below, the last column can be seen to take three values, namely 'zero', 'one' and 'two'. 'one' is assigned to clusters with more OFF feature vectors, 'two' is to clusters with more ON features and 'zero' is to clusters which are either common clusters or have very few feature vectors to decide the classification. The reason behind the low number of feature vectors is because of the shorter recordings of data for particular activities for specific patients.

If we look carefully at the table, we would see that there are rows where the total number of feature vectors are 1*. These rows of information do not provide a good response and cannot be considered as either an OFF or an ON cluster. Hence, to improve the results of the clustering process, common clusters are those clusters which have a very close call between the number of feature vectors classified as OFF and ON. In this scenario, we also include the '1*' rows into common clusters as they do not provide sufficient information in the classification process. To make the algorithm more stringent to such scenarios, any cluster with less than four feature vectors were discarded.
Table 4.2: Example of classification using K means clustering and Hard Label Method

<table>
<thead>
<tr>
<th>Total Feature Vectors</th>
<th>Ground Truth</th>
<th>OFF Classified</th>
<th>ON Classified</th>
<th>Algorithm O/P</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>1</td>
<td>15</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>14</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>17</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>15</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>3</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>5</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>6</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>5</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>1*</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1*</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1*</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1*</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>4</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>2</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>2</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>0</td>
<td>18</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>2</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>2</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>2</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>5</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>2</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>3</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>2</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>2</td>
<td>15</td>
<td>2</td>
</tr>
</tbody>
</table>
4.4.2 Fuzzy Label

Fuzzy labeling is another technique which can be used to count the vote which decides the OFF vs ON situation. In this method, classification occurs similar to the Hard labeling technique.

One of the first steps to beginning this process involves the gathering of all feature vectors related to an activity performed by a patient. These feature vectors are then fed to the clustering algorithm which then assigns each feature vector to a cluster.

The fuzzy label technique determines the label for each feature being classified with the help of a membership matrix. The size of a membership matrix can be given as the number of discriminant clusters against the number of clusters. Every value in the membership matrix corresponds to the ratio of number of feature vectors present in each cluster to the total number of feature vectors in the cluster. The concept of a scatter vector is used where each value of the scatter vector corresponds to the number of feature vectors that fall in the discriminant cluster. The product of the membership matrix and the scatter vector yields the desired classification result [8]. This leads to the completion of the classification process using fuzzy labeling technique.
Chapter 5

Results and Analysis

The results produced by the algorithm can be divided into two groups. One includes the results for the short data set; i.e. the data set which has the information when the patients were asked to do 30 sec activities. The other group involves the Patients who were asked to perform a variety of activities over a longer interval of time. The complete recording for each patient in the second data set was roughly two hours long.

5.1 Short Data Set: K means - Hard Label

The table below shows the detailed training and testing average accuracy for each patient using K means Hard labeling concept. The whole assessment process involves the algorithm to perform the analysis for ten rounds of iterations over which the average accuracy for the training and testing process is achieved.

<table>
<thead>
<tr>
<th>Patient #</th>
<th>Training Accuracy (%)</th>
<th>Testing Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>86</td>
</tr>
<tr>
<td>2</td>
<td>95</td>
<td>81</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>64</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>95</td>
</tr>
<tr>
<td>5</td>
<td>95</td>
<td>75</td>
</tr>
<tr>
<td>6</td>
<td>89</td>
<td>78</td>
</tr>
<tr>
<td>9</td>
<td>100</td>
<td>97</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>67</td>
</tr>
<tr>
<td>11</td>
<td>100</td>
<td>85</td>
</tr>
<tr>
<td>12</td>
<td>79</td>
<td>86</td>
</tr>
<tr>
<td>13</td>
<td>96</td>
<td>68</td>
</tr>
<tr>
<td>14</td>
<td>95</td>
<td>71</td>
</tr>
<tr>
<td>Average</td>
<td>96</td>
<td>79</td>
</tr>
</tbody>
</table>
This method was carried out by initializing the k means algorithm to random initial centroids at the start of every iteration; thereby increasing the different possibilities which would eventually help classify better than the previous try.

5.2 Short Data Set: K means - Fuzzy Label

The table below shows the detailed training and testing average accuracy for each patient using K means Fuzzy labeling technique. The whole assessment process involves the algorithm to perform the analysis for ten rounds of iterations over which the average accuracy for the training and testing process is achieved.

<table>
<thead>
<tr>
<th>Patient #</th>
<th>Training Accuracy (%)</th>
<th>Testing Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>92</td>
</tr>
<tr>
<td>2</td>
<td>98</td>
<td>87</td>
</tr>
<tr>
<td>3</td>
<td>99</td>
<td>66</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>96</td>
</tr>
<tr>
<td>5</td>
<td>99</td>
<td>70</td>
</tr>
<tr>
<td>6</td>
<td>83</td>
<td>84</td>
</tr>
<tr>
<td>9</td>
<td>100</td>
<td>90</td>
</tr>
<tr>
<td>10</td>
<td>75</td>
<td>67</td>
</tr>
<tr>
<td>11</td>
<td>93</td>
<td>90</td>
</tr>
<tr>
<td>12</td>
<td>74</td>
<td>83</td>
</tr>
<tr>
<td>13</td>
<td>99</td>
<td>71</td>
</tr>
<tr>
<td>14</td>
<td>100</td>
<td>68</td>
</tr>
<tr>
<td>Average</td>
<td>93</td>
<td>80</td>
</tr>
</tbody>
</table>

As we can see, this algorithm was run ten times to visualize the variation in the accuracy values of k-means hard labeling. Patient number 3, 10 and 14 have the lowest accuracy of the group of patients tested on. The clinicians provided us vital information in terms of the UPDRS scores. With the help of these scores, it is possible to determine the nature of results that can be expected from an algorithm. Due to the low variation in the UPDRS scores between the four states of the patient, it was predicted that the algorithm would struggle to carry out the classification process successfully. The lower the variation in UPDRS scores, the harder it was for clinicians to determine the transition from OFF and ON state of the patients and vice-versa. This shows us that there are still cases which cannot be tested using an algorithm due to the complexity of the disease. In general, it can be seen that the algorithm manages to classify most patients well which shows us that the tensor
decomposition technique is very helpful in extracting features and assessment of patients Parkinson’s disease.

5.3 Short Data Set: Support Vector Machine (SVM)

In this method, a 4 fold cross validation scheme was carried out to get the best results possible. The four fold cross-validation process involves the division of the training feature vectors in to four groups. Now using three fourths of the training data, a model is created with specific parameters and it tested using the remaining one fourth training data. This is carried out for every patient until it was successfully determined which patient was giving better results with a given kernel. The two types of kernel used in this method were the linear kernel and the radial basis function kernel (RBF).

With the help of the LIBSVM toolbox [3], the SVM model was created for every patient with the help of training activities and then was successfully verified with the testing activities. Following table shows the training and testing accuracy for every patient.

<table>
<thead>
<tr>
<th>Patient #</th>
<th>Training Accuracy</th>
<th>Testing Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>94</td>
</tr>
<tr>
<td>2</td>
<td>88</td>
<td>78</td>
</tr>
<tr>
<td>3</td>
<td>88</td>
<td>69</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>97</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>84</td>
</tr>
<tr>
<td>6</td>
<td>88</td>
<td>63</td>
</tr>
<tr>
<td>9</td>
<td>88</td>
<td>78</td>
</tr>
<tr>
<td>10</td>
<td>75</td>
<td>69</td>
</tr>
<tr>
<td>11</td>
<td>88</td>
<td>78</td>
</tr>
<tr>
<td>12</td>
<td>100</td>
<td>74</td>
</tr>
<tr>
<td>13</td>
<td>75</td>
<td>69</td>
</tr>
<tr>
<td>14</td>
<td>88</td>
<td>81</td>
</tr>
<tr>
<td>Average</td>
<td>90</td>
<td>78</td>
</tr>
</tbody>
</table>

With the SVM technique, several parameters were varied in order to determine the best or optimal accuracy attainable. The parameter ‘gamma’ can be defined as the inverse of the number of feature vectors for the respective patient. On performing the cross-validation method, it was determined that patients 1 and 10 were giving better results with the SVM-linear kernel whereas the other patients were giving better results with the RBF kernel.
5.4 Long Data Set: K means - Hard Label

The following table shows the results which were attained on another group of patients. Here, the patients were monitored on a longer duration (2 hours). The clinicians had given us instructions on which tasks were to be considered as training and which were to be considered as testing. With the help of the patient video log, the start and stop timings of various tasks was cataloged and used to determine the time instances for every task. The spreadsheet extracted contained recordings of two hours in length. Hence the catalog played an important role in determining which task was performed at what times.

Table 5.4: Long Data Set: K means - Hard Label

<table>
<thead>
<tr>
<th>Patient #</th>
<th>Training Accuracy (%)</th>
<th>Testing Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>86</td>
<td>82</td>
</tr>
<tr>
<td>3</td>
<td>76</td>
<td>70</td>
</tr>
<tr>
<td>5</td>
<td>98</td>
<td>97</td>
</tr>
<tr>
<td>6</td>
<td>85</td>
<td>80</td>
</tr>
<tr>
<td>7</td>
<td>86</td>
<td>75</td>
</tr>
<tr>
<td>9</td>
<td>89</td>
<td>80</td>
</tr>
<tr>
<td>Average</td>
<td>87</td>
<td>81</td>
</tr>
</tbody>
</table>

With the help of the long data set, it can be seen in the above table that the accuracy tends to increase. However, these tests were performed on another set of patients. But it is evident that the longer the signals provided, the better the results the algorithm is able to generate. With longer signals, we know that there are more feature vectors which need to be classified. This improves the algorithm in the testing phase and hence gives a better accuracy for the patient analysis.
5.5 Long Data Set: K means - Fuzzy Label

The following table shows the detailed description of the analysis which was carried out on the long data set using k means clustering and fuzzy labeling method. In this method, the k value for K means was tested with values which varied from 10-25. After carefully analyzing the results provided after every K value was executed, K was fixed at 18.

Table 5.5: Long Data Set: K means - Fuzzy Label

<table>
<thead>
<tr>
<th>Patient #</th>
<th>Training Accuracy (%)</th>
<th>Testing Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>88</td>
<td>79</td>
</tr>
<tr>
<td>3</td>
<td>75</td>
<td>66</td>
</tr>
<tr>
<td>5</td>
<td>96</td>
<td>93</td>
</tr>
<tr>
<td>6</td>
<td>83</td>
<td>77</td>
</tr>
<tr>
<td>7</td>
<td>87</td>
<td>72</td>
</tr>
<tr>
<td>9</td>
<td>87</td>
<td>78</td>
</tr>
<tr>
<td>Average</td>
<td>86</td>
<td>78</td>
</tr>
</tbody>
</table>

5.6 Long Data Set: Support Vector Machine (SVM)

The following table shows the results for the long data set. The use of the long data set helps the SVM classifier to better train itself using the larger number of feature vectors. Similar parameters were used for the long data set when compared to those used in the short data set. The four fold cross-validation technique was used in the short data set due to the small number of feature vectors. Due to the larger number of feature vectors present, the SVM classifier was trained by feeding the model with the feature vectors classified as the training activities along with their labels.

Table 5.6: Long Data Set: SVM

<table>
<thead>
<tr>
<th>Patient #</th>
<th>Training Accuracy (%)</th>
<th>Testing Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>82</td>
<td>80</td>
</tr>
<tr>
<td>3</td>
<td>80</td>
<td>69</td>
</tr>
<tr>
<td>5</td>
<td>96</td>
<td>90</td>
</tr>
<tr>
<td>6</td>
<td>87</td>
<td>74</td>
</tr>
<tr>
<td>7</td>
<td>86</td>
<td>63</td>
</tr>
<tr>
<td>9</td>
<td>92</td>
<td>83</td>
</tr>
<tr>
<td>Average</td>
<td>87</td>
<td>77</td>
</tr>
</tbody>
</table>
From these results, it can be easily said that the tensor decomposition approach to determine or assess the Parkinson’s Disease can be carried out. It can be easily said that the tensor decomposition technique was efficient in classifying most of the patients with high accuracy.

Hence the following table can be listed to summarize the accuracy which was attained in this thesis.

Table 5.7: Accuracy Summary for all adopted methods

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Method #</th>
<th>Training Accuracy (%)</th>
<th>Testing Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short</td>
<td>K means (Hard)</td>
<td>96</td>
<td>79</td>
</tr>
<tr>
<td>Short</td>
<td>K means (Fuzzy)</td>
<td>93</td>
<td>80</td>
</tr>
<tr>
<td>Short</td>
<td>SVM</td>
<td>90</td>
<td>78</td>
</tr>
<tr>
<td>Long</td>
<td>K means (Hard)</td>
<td>87</td>
<td>81</td>
</tr>
<tr>
<td>Long</td>
<td>K means (Fuzzy)</td>
<td>86</td>
<td>78</td>
</tr>
<tr>
<td>Long</td>
<td>SVM</td>
<td>87</td>
<td>77</td>
</tr>
</tbody>
</table>
Chapter 6

Conclusion and Future Works

This thesis puts together the approach of tensors along with machine learning and creates a powerful tool which can be used to carry out prediction of a particular task like the assessment of Parkinson’s Disease. This chapter gives an overall summary of the thesis, gives an insight of the findings and contributions and also throws light on the limitations of the current work.

6.1 Contribution of this Thesis

The current work has introduced the concept of tensors. Tensors can be considered as a very powerful mathematical tool which can assist users with the accumulation of related data and analyzing aspects which are hard to explore. The program developed here can help not only assess Parkinson’s disease but can also be altered to look into the signal analysis of other diseases in the bio-medical space.

Chapter 2 talks about the experimental setup and details surrounding the work stations created by the clinicians to assess Parkinson’s disease patients. It talks about two groups of people tested for varying duration of activity times. The first group of patients are asked to perform tasks for a short duration of time which spans 20 seconds to 30 seconds. The second group of patients are tested on a long time window of roughly two hours. Both groups of patients are hooked up with sensors in 6 different locations on their body. With the help of these sensors, the movements while performing the tasks are recorded and stored using the accelerometers and gyroscopes.

Chapter 3 talks about the methods which have been adopted in order to carry out the algorithmic conversion of the objective. The use of tensors can be defined as a new technique
to analyze signals. With the help of Short Time Fourier Transform, both the time domain and frequency domain can be carefully looked into. It lets us look at patterns with more clarity and determine significant features. This chapter also covers the fundamentals of the two machine learning algorithms used; K means clustering and Support Vector Machines.

In Chapter 4, the main approach towards solving the problem is discussed in the form of feature selection and extraction. In order to validate the features being used, Statistical analysis is carried out in the form of the Jarque-Bera Test (jb test), the two-sample t test and the Ranksum test. The labeling methods like Hard and Fuzzy labeling has been discussed which helps in the assessment process. It plays a vital role in providing feature vector labels when fed to the model being trained; thereby improving the model during the testing process.

6.2 Future Works

As we can see in the Results section, there certainly is scope for improvement. This section talks about the various other aspects which can be applied to better improve the data set provided to us. Following are some of the main topics which can be focused on.

6.2.1 Varying sensor combinations

As we can see, the sensors being used in this thesis involves the wrist, trunk/thigh and the ankle/foot. The data worked on by these sensors used is close to half the data provided. By using the head sensor and arm sensor and creating larger multi-dimensional tensors, more features could be extracted which could aid the assessment process. Also using all the sensor information provided can be accommodated using tensors.

6.2.2 Accelerometer Signals

In this thesis, the gyroscope signals have been used extensively. By carefully analyzing the accelerometer signals, one can develop the same method to determine the assessment process which would unfold. The results of the accelerometer and gyroscope could be clubbed in order to see how the results turn out. It would be interesting to see the manner in which the accelerometer contributes to the assessment process.
6.2.3 Other machine learning algorithms

Having covered k means clustering and support vector machines in this thesis, the possibility of using Self Organizing Tree Maps (SOTM), K nearest neighbors (KNN), etc. would help understand and validate the results produced in this thesis. The fact that machine learning algorithms exist in such large variations, it would be an interesting idea to experiment and analyze the signal data provided by the clinicians.

6.2.4 Tensor Decomposition Techniques

Canonical Polyadic Decomposition has been carried out throughout this thesis and very minimal light has been shed on LMLRA and BTD. These two techniques are based on the concept of factor matrices and core tensors. The presence of core tensors has not been covered in this thesis but would provide another area of research which can be looked into.
Bibliography


