HAND GESTURE RECOGNITION USING UNSUPERVISED LEARNING

A Thesis submitted to the faculty of
San Francisco State University
In partial fulfillment of
the requirements for
the Degree

Master of Science

In

Engineering: Embedded Electrical and Computer Systems

by

Sushmalekha Shankar Birur

San Francisco, California

May 2018

CERTIFICATION OF APPROVAL

I certify that I have read Hand Gesture Recognition using Unsupervised Learning by Sushmalekha Shankar Birur, and that in my opinion this work meets the criteria for approving a thesis submitted in partial fulfillment of the requirement for the degree Master of Science in Science: Embedded Electrical and Computer Systems at San Francisco State University.

Dr. Xiaorong Zhang, Ph.D.
Assistant Professor

Dr. Hamid Mahmoodi, Ph.D.
Professor

Hand Gesture Recognition Using Unsupervised Learning

Sushmalekha Shankar Birur
San Francisco, California
2018

Gesture recognition is one of the revolutionary technological advancements seen in human-computer interaction field today. The applications of gesture recognition are vast from sign language recognition, through prosthesis control, to control interface for virtual reality (VR) and augmented reality (AR) systems. Many of these applications use supervised learning-based machine learning algorithms to interpret data and make decisions.

Unsupervised learning is still in the early development stage since it involves streaming in continuous unlabeled data sets and obtaining meaningful data sets out of billions of data remains a challenge. This project aims to develop and analyze k-means clustering - an unsupervised learning technique, for hand gesture recognition based on unlabeled electromyographic (EMG) data sets obtained from a commercial armband Myo developed by Thalmic Labs.

I certify that the Abstract is a correct representation of the content of this thesis.

Chair, Thesis Committee

5/21/2018
Date

ACKNOWLEDGEMENTS

My special thanks to my thesis advisor Dr. Xiaorong Zhang, School of Engineering at SFSU. Dr. Zhang's office door was always open whenever I ran into any trouble and suggestions needed. She has given me complete freedom to write this paper based on the ideas I came up with and always steered me in the right direction. I like to thank my thesis committee member Dr. Hamid Mahmoodi, School of Engineering at San Francisco State University. Dr. Mahmoodi has been a mentor throughout my graduate coursework and has been a great support. My humble thanks to Dr. Christopher Pong, Director of School of Engineering at SFSU. He has been my supervisor as well as my mentor throughout the journey of my graduate studies.

I would like to express my gratitude to my beloved mother Sunanda S for providing me with unfailing support and continuous encouragement throughout my years of study and the process of writing this thesis. This accomplishment wouldn't have been possible without her support. I thank my friends Ajay, Ragh, Ashok, Avinash, Rakesh, Ambica, Vyshakh and Sahana for a memorable grad life and continuous support in academic accomplishments. I would like to thank my colleagues Krithika and Mike for all the support at Molecular Devices. Finally my entire work is dedicated to my father Late Mr. B. V. Shankar.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figures Page

Chapter 1
Introduction

Gestures are part of our everyday lives. While performing day to day activities, talking to people, or just using the cellphones, gestures become necessary for communication or to imparting meaningful information. Any meaningful motions like movement of eyes, face, head, arms or body are considered to be gestures. In brief gesture recognition can be termed as a process in which the gestures performed by a user are recognized by the receiver.

Gesture recognition is a computing technique that helps in gathering meaningful data via human body motions and makes meaningful sense out of it. In recent years gesture recognition has been used in areas of human-computer interaction, entertainment, and healthcare and so on. This includes, typical applications like sign language recognition, robot surgery, patient health monitoring systems, gaming consoles.

Human Computer Interaction (HCI) technology which focuses on studying the interaction between human beings and computers has become one of the revolutionary technologies. Gesture recognition has become a significant sub space in HCI with increasing demand in developing intelligent systems that can translate human motions into computer recognizable applications and vice-versa.

1.1 Applications of Gesture Recognition

The applications of gesture recognition are vast from sign language recognition to more sophisticated applications like robotic surgery. Hand gesture recognition is one of the key parts of human activity recognition and plays a vital role using gesture recognition. The major applications of hand gesture recognition in the recent days are used in biomedical applications like training robots for human assistance for the old and disabled, robot assisted surgeries, sign language recognition, prosthetics development, patient monitoring system, gesture to speech, virtual environments, 3D modeling, television control, multimodal interaction[13][19][21].

The applications of gesture recognition can be extended to the gaming industry as well. With the help of hand gestures and AR/VR devices, games today provide experiences close to real time, live and human-like. Typical AR/VR devices are like Oculus VR headsets. AR/VR devices applications extends to education, as trainee's can be put through situations they would never otherwise get to experience, such as in-flight simulators.

Internet of Things is an upcoming automation tech that can change the mode of operation of work and household. With hand gestures, household appliances can be controlled, enabling smart homes and with IoT, devices can be linked to perform multiple operations, which otherwise would require human intervention and the time taken is also a key factor.

Human-robot interaction [1] is another area where hand gesture recognition has been successfully used. The use of keyboard and mouse is limited to 2D world, but the controlling of a robot is in 3D space. Hand gesture is most suitable for such purposes. Simple commands are being used for the control of the robot, such as signs, numbers, actions etc.

Similarly, 3D CAD modeling inputs are provided by hand gestures. The 3-draw technology developed by MIT [2], is a pen embedded in polhemus device to track the position and orientation of the pen in 3D. A 3D space sensor is embedded in a flat palette that represents the plane of the object. The CAD model is moved synchronously with the user's gesture and objects can thus be rotated and translated in order to view them from all sides as they are being created and altered.

Realizing the importance of hand gesture recognition and its key importance in the above mentioned applications, our study is tailored to recognize activity using hand gestures.

1.2 Activity Recognition

For more than a decade now, Human Activity Recognition (HAR) is a key area of research and takes a complete new perspective when concentrated towards wearable sensor computation. This section gives an audit about HAR based on wearable sensors.

HAR based on wearable sensors faces many issues that motivate development of new technique to achieve high accuracy systems based on more realistic situations. Some of the key challenges associated are:

1. Attribute selection for measurement

2. Developing non-intrusive monitoring systems, high portable and inexpensive systems

3. Design of feature extraction and inference methodology

4. Data collection in realistic world

5. Supporting new users for flexibility without re-training needs

6. Implementation on mobile devices that meets energy and processing requirements

Two main approaches are widely seen for HAR

1. Using external sensors
2. Using wearable sensors

Using external sensors on platforms like home-monitoring systems are known for recognizing complex tasks like eating, taking a shower, washing dishes which requires number of sensors to be placed in a definitive region that needs interaction with the subjects. Once the subject is out of sensors' range then such a system will be unable to collect data and infer any relevant information.

External systems are expensive because of installation and maintenance of sensors. Additionally, securing the home-monitoring systems which includes cameras invades privacy and inferring the activities from video sequences are computationally expensive which hinders the HAR scalability.

The above mentioned limitations accelerate the use of wearable sensors for HAR. The types of attributes measured are user's movement, environmental variables and physiological signals that includes accelerometers, GPS, temperature, humidity, heart rate. Electromyography (EMG) Sensors are one of those devices which help in analyzing the bioelectric signals and measuring the electrical activity of the muscle while at rest and while in contraction of hand movements by humans. EMG is used in diagnosing many neuromuscular diseases, kinesiology; diagnose functional abnormalities in the muscles and other motor control related disorders. They are also used as a control signal in prosthetic hands, arms, lower limbs and other prosthetic devices. They can measure electrical signals, filter and rectify electrical activities as well as predict the muscular movement as necessary.

Our study focuses on measuring and using electromyography (EMG) signals from wearable sensors incorporated in myo armband used by the subject.

1.3 Technologies used in activity Recognition

Dorra Trabelsi, Samer Mohammed and Yacine Amirat [4] work particularly focuses activity recognition using body mounted sensors explaining the advantages that are overcome using a vision based system given the fact the system has to be used in a confined space with controlled environmental parameters without intruding personal privacy and difficulty of calibrating the cameras.

Body mounted sensors have shown promising results measuring human activities in laboratory as well as free living environment along with worthwhile mentioning of advancement in MEMS technology.

From this work, the daily activities can be clearly classified into

1. Static Type
2. Dynamic Type

Static activities include standing, sitting and lying, whereas dynamic activities include walking, running, jumping. To recognize these two types of activity classes, supervised and unsupervised classification has been used in many works. Trabelsi's work

[4] concentrates on unsupervised learning which does not require labeled data sets that overcomes the limitations of supervised learning techniques for free living spaces.

Unsupervised recognition models are classified into two categories

1. Static classification approaches
2. Temporal classification approaches

Typical static classifiers include mainly k-Means, DBSCAN (Density-Based Spatial Clustering of Applications with Noise). Considering Trabelsi's work [4] as a base, our research work focuses on applying k-Means classification for activity recognition and activity being more focused to hand gestures.

Chapter 2

Supervised and Unsupervised Learning

2.1 Supervised Learning

As discussed in Section 1.3 several methods are used to collect data for HAR and often these raw data are useless if further effort of knowledge discovery through context awareness is not applied. HAR systems make use of machine learning tools to analyze the underlying context obtained from the raw data to recognize pattern for analysis and predict the data [22].

In machine learning, recognition of pattern from a set of observations forms instances. The input set of observations is called training set. The training set may or may not contain labels i.e., walking, running, standing etc.

Based on labeled training set, machine learning has two approaches

1. Supervised Learning
2. Unsupervised Learning

Over the decades, a HAR system is used to return a label from the training set and has made use of supervised learning. As seen in Fig.1, the input training set is obtained

under the supervision of research team and manually specifies the activity labels, timestamps and stores the data.

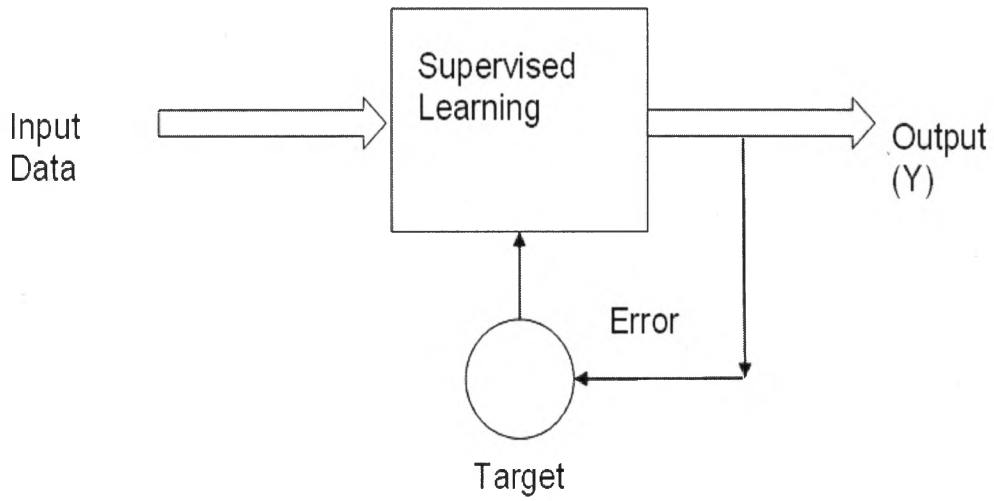

Fig1. Model representation of supervised learning. Image retrieved from [8].

The supervised learning models such as Decision trees, Support Vector Machine (SVM), Bayesian methods are applied to the labelled training set. The fact that the method is called Supervised learning because, the method acts like a teacher to supervise the learning process until a satisfactory output is obtained by iteratively correcting the errors from the process each time. Supervised learning can also be represented in equation form as below:

$$Y=f(X) \quad (1.1)$$

Where, Y is the output variable, X is the input variable, f () is the function that maps output to input.

Per above equation, supervised learning can be defined as a process that applies an algorithm to learn the mapping function from input to the output. Having learnt the basics of supervised learning model, let us survey some of the popular supervised learning methodologies which mainly are classification models

a. Decision Trees

Decision trees form a hierarchical model that has nodes mapped to the features and the edges represent the possible feature values. Each branch derived from the root node is based on a classification rule. C4.5 is the most widely used decision tree algorithm. Associated with each training case is a label representing the name of a class. Each internal node of a decision tree contains a test, the result of which is used to decide what branch to follow from that node. In classification mode, when a test case (which has no label) reaches a leaf node, C4.5 classifies it using the label stored there.

b. Bayesian

This classifier method calculates posterior probabilities for each class using estimated conditional probabilities from the training set. Bayesian Network classifier and Naive Bayes are the widely used Bayesian classifiers. The main issue this classifier suffers from is

assuming all features are independent for a given class value. Such assumption doesn't hold good in HAR which measure ECG, acceleration values. Hence a topology construction using Bayesian for our study is obsolete.

c. Instance Based Learning

The methodology classifies similar instances by comparing with each data with the whole training set during training phase. Such method becomes computationally expensive and requires more storage, which cannot be implemented for mobile devices

d. Support Vector Machine and Artificial Neural Network

SVMs depend on kernel functions that estimate all instances to a hyper plane with the intention of finding a linear decision boundary to divide the data. Whereas Neural Networks mimics the human brain's neuron behavior spreading activation signals and encoding knowledge in the network links. Hence ANNs are associated with high computational cost and requires large amount of training data.

2.2 Advantages of Supervised Learning

1. Supervised learning are best suitable for classification of the data hence it has gained the popularity in HAR. Important usage of Supervised learning is used for Classification problem i.e., given a set of input, the model can place the data to the right category at the output

2. Supervised learning method always has input - output pair to train the machine during training phase.

3. Supervised learning model are widely adopted for making predictions for continuous quantity called as Regression learning previous examples

2.3 Disadvantages of Supervised Learning

1. The model always has to be fed with input-output pair for the model to either predict or categorize the data. In order to produce the input-output pair of data, the data collection protocol involves manual supervision and an expert to label the input data correctly

2. The process of labelling becomes laborious when the data grows large

3. The model cannot recognize the underlying structure of the data and cohesion of the data unless a labeled data set is provided to the system

4. Aforementioned popular supervised learning algorithm proves that they become computationally expensive when it comes to training the system with large data sets and storage issues for mobile applications

2.4 Unsupervised Learning

Unsupervised learning consists of only input data (X) and the output is unknown. The system does not have any teacher to monitor the resulting output, hence the name Unsupervised. The system is allowed to discover the underlying structure of the data set provided as input which operates on the basis of grouping the data that has similar feature sets.

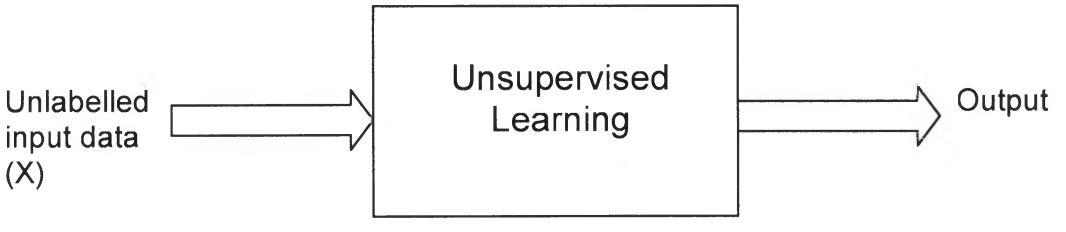

Fig2. Model representation of unsupervised learning. Image retrieved from [8].

Main difference that can be inferred from Fig 1 to Fig 2 is that Unsupervised learning is able to eliminate the labelled input data to the system and overhead cost of retraining cost by feeding the error back to train the system.

The most common unsupervised learning algorithms are clustering, principal component analysis (PCA) and Independent Component Analysis (ICA).

a. Clustering

Clustering is an assignment operation on a set of objects such that objects in the same group (cluster) are more similar to each other than to those in other groups. Clustering algorithm is further divided into Centroid-based algorithms, Connectivity-based algorithms, Density-based algorithms, Probabilistic, Dimensionality Reduction and Neural networks / Deep Learning. Our study mainly focuses on Centroid-based algorithm. Centroid-based clustering is a static classifier which employs k-Means. Further details of k-Means are explained in the methodology section of our work.

b. PCA

It is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. Few of the applications of PCA are compression, simplifying data for easier learning, visualization.

c. ICA

It is a statistical technique used to figure out the hidden factors that underlie sets of random variables, measurements, or signals. ICA defines a generative model for the observed multivariate data, which is typically

given as a large database of samples. In the model, the data variables are assumed to be linear mixtures of some unknown latent variables, and the mixing system is also unknown. The latent variables are assumed non-Gaussian and mutually independent and they are called independent components of the observed data.

2.5 Advantages of Unsupervised Learning

1. The system eliminates the manual intervention to supervise the data collection and data labelling for large set of data.
2. The model doesn't have to be fed with input-output data pair to teach the system to produce an output.
3. By recognizing the underlying structure in the input data having feature similarities, the system reduces the data pre-processing time during training phase.
4. It relies on statistical properties of the data.
5. Typical application of this system is associated with problem statement that involves clustering analysis and association type of problems.

2.6 Disadvantages of Unsupervised Learning

1. Since the system is not trained with input-output data pair, there is no concept of correct output obtained from this system

2. Unsupervised learning are often defined to be complex since the system determines the output from the underlying structure with the input data

3. Sensitive to scale: rescaling the datasets i.e., normalization or standardization would completely change results which may add extra time

Chapter 3

EMG based Gesture Recognition

3.1 Electromyography (EMG)

Electromyography (EMG) is an experimental technique concerned with the development, recording and analysis of myoelectric signals. Myoelectric signals are formed by physiological variations in the state of muscle fiber membranes. Applications of EMG are mainly seen in Medical Research, Rehabilitation, Ergonomics and Sports Science. Our study focuses on application of EMG for movement analysis related to Sports Science.

Fig 3. Myo armband. Image retrieved from [10].

From Fig 4, a smallest functional unit to describe the neural control of the muscular contraction process is called a Motor Unit. It is defined as the cell body and dendrites of a motor neuron, the multiple branches of its axon, and the muscle fibers that innervates it. The term units outlines the behavior, that all muscle fibers of a given motor unit act "as one" within the innervation process.

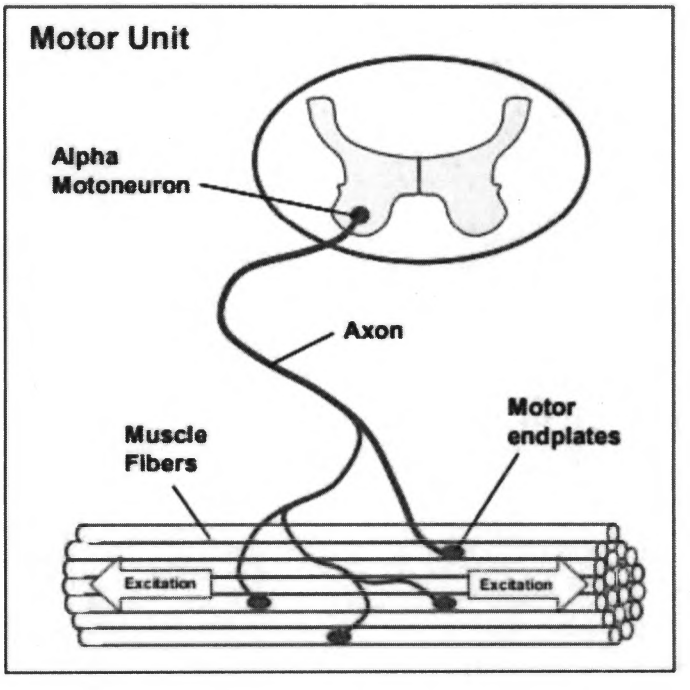

Fig4. Single motor unit. Image retrieved from [11]

The EMG - signal is based upon action potentials at the muscle fiber membrane resulting from depolarization and repolarization processes. In Fig 5, Superposition of Motor Unit Action Potential (MUAPs) within kinesiological studies the motor unit action potentials of all active motor units detectable under the electrode site are electrically

superposed and observed as a bipolar signal with symmetric distribution of positive and negative amplitudes (mean value equals to zero). It is called an Interference pattern.

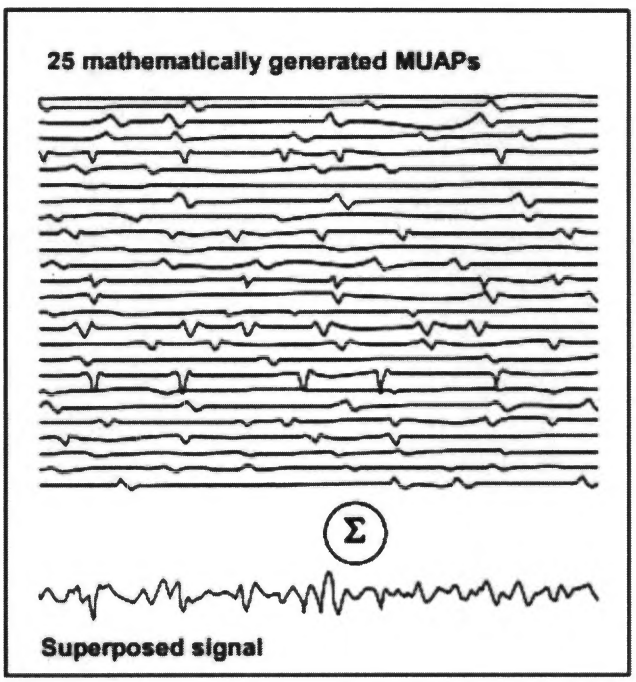

Fig5. Superposition of MUAPs to a resulting electromyogram. Image retrieved from [11].

An unfiltered and unprocessed signal detecting the superposed MUAPs is called a raw EMG Signal. In the example given below in Fig 6, a raw surface EMG recording (sEMG) was done for three static contractions of the biceps brachii muscle. When the muscle is relaxed, a more or less noise-free EMG Baseline can be seen.

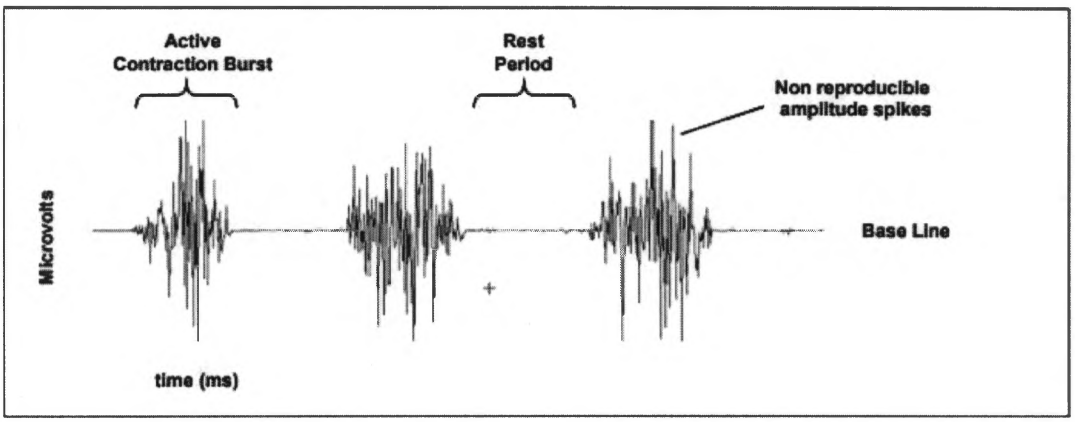

Fig 6. Raw EMG signal for three static contractions.Image retrieved from [11].

Electromyography (EMG) is a diagnostic procedure to assess the health of muscles and the nerve cells that control them (motor neurons). The Myo armband also has a nine-axis inertial measurement unit (IMU) which contains a three axis gyroscope, three axis accelerometer, three axis magnetometer and houses eight EMG sensors. From these units, the orientation and movement of a wearer's arm can be determined through analyzing the spatial data acquired.

3.2 Related work

Previous work by A. Nguyen explains the advantages of clustering algorithm that, there is no need for training/development data, no threshold adjustment requirements, and that it is robust to different data conditions [5]. This publishing proposed novel approach for fall detection by applying Expectation Maximization (EM) for usual daily activities of

the subjects. Maximum a posteriori (MAP) was applied for adaptation i.e., unusual event clustering adapted from usual events.

The coherent clustering algorithm's hyper parameters, namely, the number of initial clusters (K), number of Gaussian components (M), minimum duration constraint (MD) and the set of features used for cluster discrimination were required to be set [5]. Average cluster purity (ACP) and Average event purity (AEP) were used to measure cluster performance and scores for the clustering were as high as 0.84 and 0.79 for ACP and AEP, respectively, with the best overall purity score of 0.81, which was achieved using the raw feature.

Even though these results of unsupervised clustering were promising to provide great segmentation outputs of usual and unusual events, it was not able to recognize the type of activity.

Stephen O'Hara, Yui Man Lui and Bruce A. Draper [6] make use of Bag of Features and Product Manifolds for clustering video clips of human facial expressions, hand gestures, and full-body actions. They show that PM yields superior results when measuring the alignment between the generated clusters over a range of K-values (number of clusters) and the nominal class labeling of the data set. A key result is that unsupervised clustering with PM yields accuracy comparable to state-of-the-art supervised classification methods.

Chapter 4

K Means Algorithm

In this section, the general definition of K-means is represented, with formulae for better understanding [7]. K-means algorithm is a kind of hard clustering algorithm, which means that it divides n data samples of n-dimensional Euclidean space into K classes. The accurate clustering number K is determined, and K objects are randomly selected as center points.

Once the number of clusters k is determined according to the procedure, K centroids are chosen from the dataset, in random, C_1, C_2 .... C_k. Let the dataset be defined as

$$X = x_1, x_2, x_3, \ldots\ldots x_n \quad (4.1)$$

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (||x_j - \mu_i||^2) \quad (4.2)$$

Where, each data x1, x2, etc. are of n-dimension.

The centroids are chosen based on within-cluster sum of squared criterion. The closest C_k of each data point x_n is taken as the centroid of the cluster to which x_n belongs. This above process is repeated for all data points.

Once the clusters are formed, new centroids are calculated by computing the centroid of each cluster. This is done by computing the average of each dimension of the data point within a cluster. The value of shift between the old and the new centroid is computed.

The above iteration is carried until the value function converges or min. shift is computed. Inertia has one of the drawbacks of not being a normalized metric. In very high dimensional spaces, Euclidean distances tend to become inflated. Dimensionality reductions algorithm like normalization can help resolve such drawback.

4.1 Determining K

In K-means, a dataset comprising of elements from n-dimensional vector space is clustered into k classes. The challenge before the process of clustering is determining the no. of clusters, k. In each cluster the volume is defined as the product of the range of cluster in each dimension.

For a two-dimensional dataset, like points in a 2-D Cartesian plane, the 'volume' of each cluster is equal to the area of the smallest square (aligned with the axes) that can completely contain all points in the cluster. For a three-dimensional dataset, the 'volume' of a cluster is equal to the volume of the smallest cube (aligned with the axes) that can contain all points in the cluster [8].

The algorithm to determine the volume and number of clusters can be summarized as follows:

1. Volume of Clusters
    - Consider a cluster as k
    - From the vectors within the cluster, determine the max value of each dimension within the cluster, $V_{i,max}$, where 'i' is the dimension
    - Determine the min value of each dimension of the vector within the cluster, $V_{i,min}$
    - Calculate the range, $R_i = V_{i,max} - V_{i,min}$
    - The product of R_i for all dimensions, i=1, 2…n, of the n dimensional vectors provide the volume
    - Repeat the above for every cluster
2. Optimal number of clusters k
    - Start with K=1 as the optimal number of clusters
    - Calculate the volume of the cluster, V_1
    - Increment k by 1
    - Perform K means clustering to obtain k clusters
    - Calculate the ratio V_k/V_{k-1}. V_k is the sum of volume of k clusters
    - If the above value is less than a threshold, repeat from step 3 Else, proceed
    - The value k-1 is the optimal number of clusters.

This method relies on the fact that creating k+1th cluster after reaching the optimal number of clusters would be overlapping and redundant in volume. This method is a variation of the elbow method suggested by Thorndike (1953), which relies on variance. Volume is chosen for better visualization if required.

4.2 Cluster density

Cluster density is an analysis parameter i.e. is used for keeping record of the cluster and its properties as part of analysis. Typically, it is calculated by considering the variance and standard deviation of a cluster and is reflected to every element of the data.

The general computation pipeline would be -

- Compute the radius of the circle of influence, r_o around a centroid
- The radius would be the average variance of the cluster. The reason this is chosen is to eliminate outliers
- Compute the area of the circle using radius r_o, A_o
- Compute the volume of an encompassing sphere using the same radius, V_o
- The density can be defined as $D_o = V_o/A_o$

Chapter 5

High level architecture

In our current study we propose a novel approach of unsupervised clustering to determine the type of activity performed using Myo Armband. This could result in evaluation of clustering for appropriate data segmentation and overcome the drawback of recognition of activity.

Predictive model of unsupervised learning for clustering in our work is represented as shown in Figure 7.

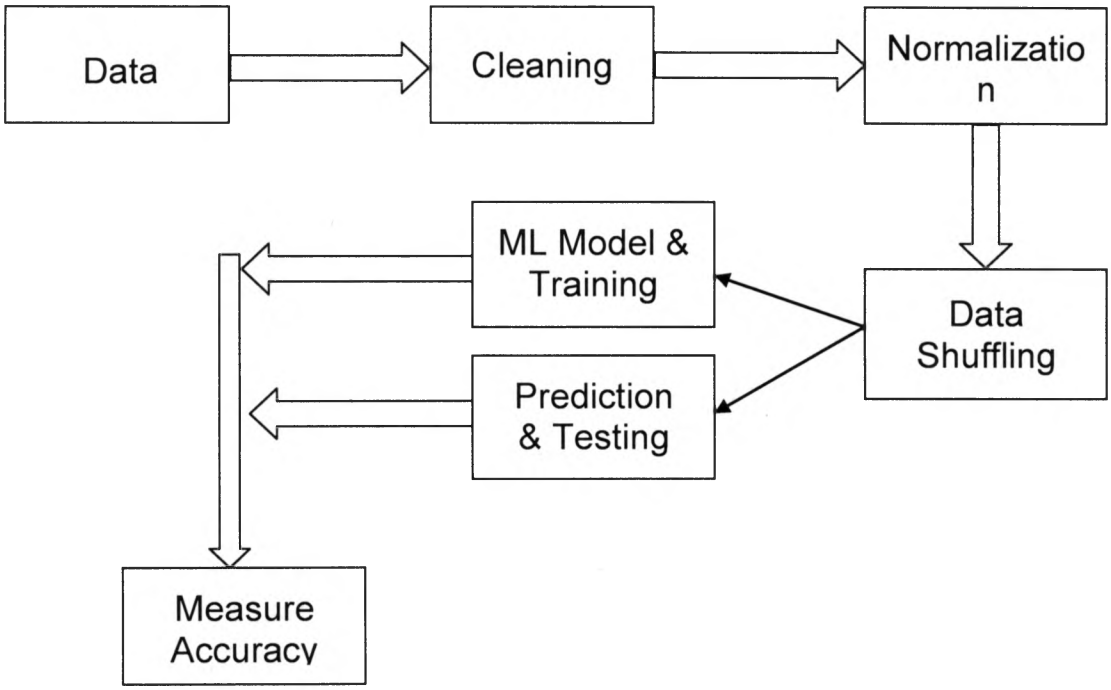

Figure 7: High level architecture of unsupervised learning predictive model representation

From Figure 7, High Level architecture provides a bird's eye view of all the essential components of the entire project setup. It helps in identifying and interfacing the entire system that would be used for development of product or for experimentation purposes.

Data collection is the first phase in which Myo armband is used to capture raw EMG values from 8 EMG sensors for 5 different gesture sets. Each gesture's EMG values is captured for a time period of 6 seconds and each gesture is repeated 10 times.

Total data collection is performed for time period of 300 seconds. The data is stored in the form of comma separated file (csv) format.

The data collected is cleaned for any irrelevant data obtained due to noise. Using .csv format to store the data one of the common issues detected is inclusion of white spaces. We adopt data cleaning process to eliminate any empty spaces in the .csv format using .dropna() function using python. Data cleaning is following by Normalization to eliminate any data redundancy and vectorize the input data into a scalable range that will improve the overall results. Data cleaning and normalization are adopted to produce quality input data for model training.

Data shuffling is performed to split the data set to training and testing set. In our model, we are splitting the data set in 70% as training and 30% as testing set with a random seed of 123. This evaluates the quality of the model by covering major portion of the data in to training and measure the accuracy of the model on the testing set.

Machine learning model training is performed using k-means clustering. The implementation of k-means clustering is explained in detailed in Chapter 6. The trained model is evaluated for testing data set and accuracy of the model is measured using Silhouette scoring and elbow method.

Chapter 6

Implementation

6.1 Hardware Setup

For the study of unsupervised learning of hand gesture recognition, we are using Myo armband by Thalmic labs which is a gesture recognition device. This device uses EMG sensors which help in sensing electrical activity in the forearms. This device is made up of Gyroscope, Accelerometer and Magnetometer which helps in measuring axis/direction, acceleration forces and magnetic forces respectively.

Thalmic Labs are one of the pioneers in the HCI industry which develops wearable technology solutions. Myo armbands when worn on the forearms, will allow users to control devices wirelessly using hand motions. This device is used in various applications - while playing video games, visual entertainment appliances and music; just to mention few. The various values from the sensors mentioned above which are in the myo band could be captured by the raw sensor values which could further be used for scientific analysis like soft computing technology like machine learning to obtain meaningful data to detect different hand gestures

The technical specifications of myo band that is used for our study includes

1. Myo Armband
2. Micro-USB cable which is used as a power supply for Myo band
3. Bluetooth adapter which is use for connecting to the computer to capture the sensor values
4. Myo sizing clip used for adjusting the armband size to fit the forearm

This Myo Sensor made up of Medical grade Stainless steel EMG sensors, high sensitive 9 axis IMU with 3 axis gyroscope, 3 axis accelerometer and 3 axis magnetometer. This band also has a ARM cortex M4 processor.

6.2 Software Setup

For the study, software setup uses PC with Ubuntu 16.04 - 64bit with 7 GB RAM with Intel i7 2Ghz processor . I am using Python v2.7 programming language for development and analysis of unsupervised learning of hand gesture recognition. Also, I am using Anaconda Navigator - a desktop GUI tool for managing conda packages, environments and channels to avoid using command line commands.

Python being one of the flexible languages, under OSI license approved open source programming language has gained an immense popularity among data science folks. This provides us with various special packages, libraries and implementations for solving and implementation of machine learning techniques.

6.3 Python Packages

Scikit-Learn python package is used in this study. This package is contains a large set of machine learning - both supervised and unsupervised learning algorithms with a clean interfacing. This package was built upon SciPy and was developed by David Cournapeau [23].

NumPy, a primary package for scientific computing is a part of Scikit-Learn package. This package can handle humongous multidimensional arrays, contains large collection of mathematical methodologies, linear algebra / Fourier transformation formulations which can be applied on huge arrays and rapid calculations.

Matplot and Pandas also a part of Scikit-Learn packages are used in the development which are used for 2D/3D plotting and Data Structure Analysis respectively.

6.4 Algorithms used

In the current study for data analysis and machine learning implementation the different algorithms used are L2 Normalization, K-Means, Silhouette analysis and Elbow method.

L2 Normalization is used for preprocessing the dataset. Data Normalization improves the quality of data set by reducing and eliminating the data redundancy. Also

better clustering and improved performance is achieved with the preprocessed dataset, especially for k-means algorithm

K-Means is the fundamental and most popular clustering technique used in unsupervised learning. Clustering technique attempts to learn from the properties of data set for discrete labeling of group of points. This property of k-means is at most importance in our study since our focus is on unlabeled data sets. In our study k-means is used for training the model.

Silhouette analysis generates silhouette score which helps in measuring similar objects to its own clusters compared to peer clusters. Higher the score, better the match to its own cluster and poorly matched to its peer cultures.

Elbow method is used in finding out the optimal number of clusters for k-means clustering. This method helps to determine the right number of cluster that could be used for k-means algorithm for the user.

Chapter 7

Experiment setup

The datasets comprises of six different gestures. The gestures are static and held in rest position with subjects held down on table seated relaxed on chair in front of laptop setup on the table, see **Fig 8 Gesture 1. Table 1** shows description of each gesture.

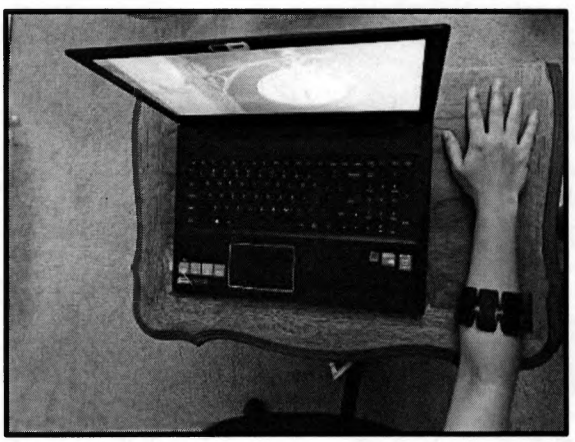

Fig8. Gesture 1: Hand in rest position (comfort)

The rest position in our method is considered as a reference (works as a base) and the remaining four gestures are chosen such that each directional and orientation perimeter is at its extreme boundary values.

We can derive our gesture selection performed by the subject for Gesture 1 as "Comfort" and Gestures 2 to 5 as "Extreme" i.e., Gesture 2 is extreme orientation turn

position by the subject turned towards left, Gesture 3 is extreme wave out position for the wrist by the subject turned towards right, Gesture 4 is extreme wave in position for the wrist by the subject turned towards left and Gesture 5 is extreme pressure applied to form a fist.

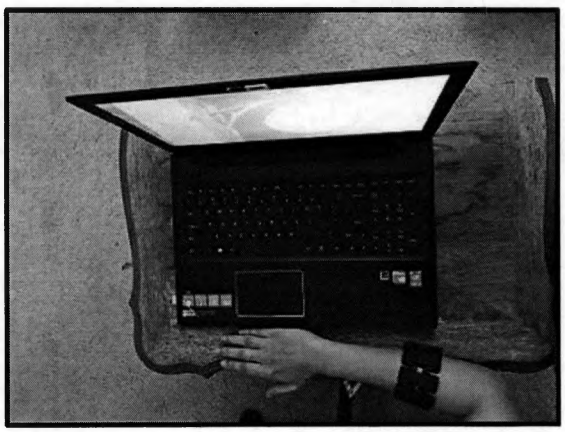

Fig9. Gesture 2: extreme orientation turn position by the subject turned towards left

Fig10. Gesture 3: extreme wave out position for the wrist by the subject turned towards right

Fig11. Gesture 4: extreme wave in position for the wrist by the subject turned towards left

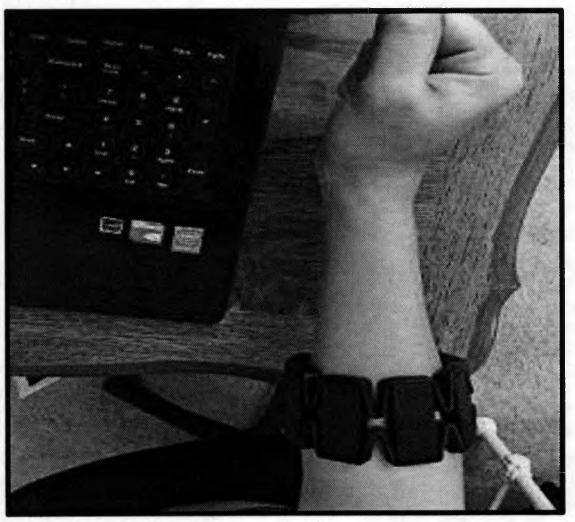

Fig12. Gesture 5: extreme pressure applied to form a fist.

Gesture 1	Gesture 2	Gesture 3	Gesture 4	Gesture 5
Hand in rest position (comfort)	Extreme orientation turn position by the subject turned towards left	Extreme wave out position for the wrist by the subject turned towards right	Extreme wave in position for the wrist by the subject turned towards left	Extreme pressure applied to form a fist

Table 1: Five gestures performed in the current study

Gestures are performed by three different subjects, 1 female aged 25 years, 2 male aged 22 years and 25 years respectively collectively to form data sets. Each trial for a gesture is captured as explained below and the subject is instructed to perform Gesture 1 to have a comfort in their daily work and Gesture 2 to 6 are instructed to perform the most extreme bendable position that's possible to achieve wearing the armband:

A subject wave out the wrist such that the myo armband is connected to read data which is the handshake signal recognized as per the code and once the myo armband vibrates twice then it is ready and sending out the sensor values. Once the indicates the detection with the 2 times vibration the subject holds the arm in the instructed position until a timeout signal occurs to end reading values from sensors. The timeout signal is sent after 6 seconds and hence the window length for each sample is maintained constant

as 6 seconds. Each gesture is performed 10 times for a 6sec window length. Hence total number of samples:

Total no. of samples = Number of samples for each gesture * Total number of gestures

In this experiment the total number of samples are 10 * 5 = 50 which is the total number of samples in csv format obtained.

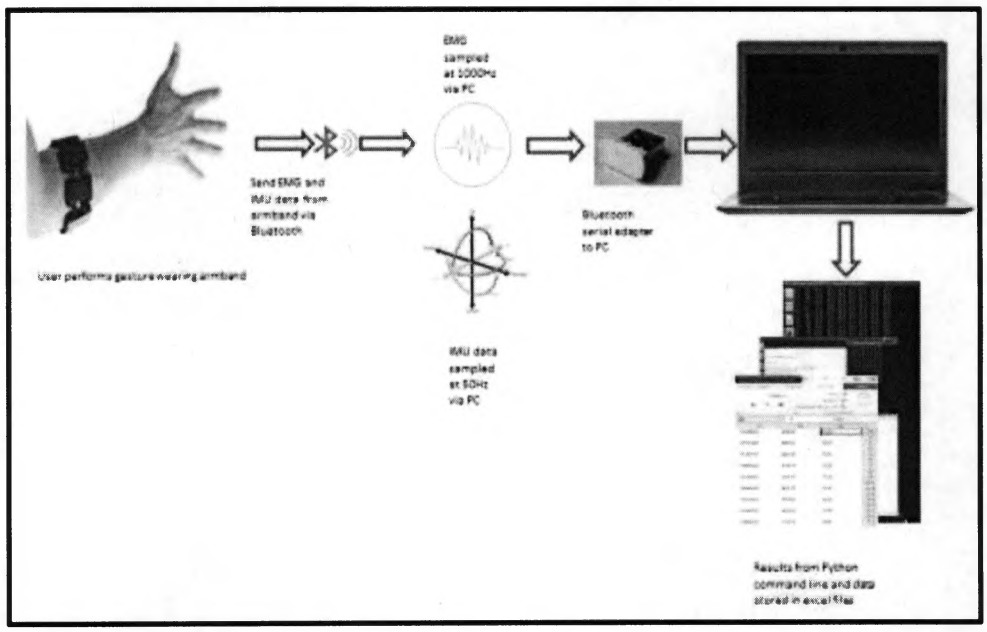

Fig 13 : Experimental setup for Myo ArmBand

7.1 Creating dataset

The data is collected using bluetooth interfacing and sampling of EMG data. The armband is connected to a server/pc via Bluetooth. The interface script is done using python and its related packages. In Fig 14, EMG frequency is set to 50 Hz.

For USB driver communication, a linux based serial tty protocol is implemented in the code. In Fig 15, ser.Serial open port command in python, specifies port at 9600 baudrate, dsrdtr=Enable hardware (DSR/DTR) flow control, DTR-Data transfer ready, DSR-data set ready.

The sensor parameters are set from the code and communicated back to the device. This is for fine tuning the sensors to detect the gesture based tendon vibrations, through their inbuilt potentiometer. An initial handshake signal is communicated from the band to pc, to setup the signal strength, data type and the noise models i.e., detecting tty as a dongle connection

The orientation data indicates the positioning of the armband in terms of roll, pitch and yaw. The angular velocity of the armband is provided in a vector format and the accelerometer represents the acceleration the Myo armband is undergoing at a given time.

Currently, the Myo armband is able to pull EMG data at a sample rate of 50Hz. Raw data of EMG is measured as an activation and is in millivolts measurement. EMG

data is obtained as 8 bit unsigned integers which in our data the range of EMG values obtained are from 0 to 255.

```
## Sampling rate of the underlying EMG sensor, capped to 1000. If it's
## less than 1000, emg_hz is correct. If it is greater, the actual
## framerate starts dropping inversely. Also, if this is much less than
## 1000, EMG data becomes slower to respond to changes. In conclusion,
## 1000 is probably a good value.
C = 1000
emg_hz = 50
## strength of low-pass filtering of EMG data
emg_smooth = 100

imu_hz = 50
```

Fig14. EMG Sampling frequency code

```
class BT(object):
    '''Implements the non-Myo-specific details of the Bluetooth protocol.'''
    def __init__(self, tty):
        self.ser = serial.Serial(port=tty, baudrate=9600, dsrdtr=1)#specificy
        self.buf = []
        self.lock = threading.Lock()#
        self.handlers = []
```

Fig15. Bluetooth object protocol for serial tty

The Myo armband is used at the widest part of the forearm, that is, the upper forearm. Sizing clips are available which allow for a more constrained grip, better suited for smaller arms. Unlike other EMG sensors, the Myo armband does not require the wearer to shave the area around which the armband will be worn. This allows for easier setup procedures in experimental or real-world environments.

The advantage that the Myo armband technology would provide is a much lower cost of purchase compared to other sEMG detecting hardware.

Data is read from Dataset.csv with the help of pandas. The first column data is extracted and converted into an array, from both the csv's.

```
    with open('emgfold1.csv','a') as csvfile:
writer=csv.writer(csvfile)
writer.writerow(emg)
#writer.writerow('\n')
    csvfile.close()
```

Fig16. Data stored to csv file

Scikit framework in Python has a rich computational and machine learning library supported by Numpy, matplotlib. The current study utilizes the features useful through python scikit tool for data analysis.

- "def proc_emg" is the function built to write data to csv files
- "open" is the in-built function used to create a csv file with two parameters, one specifies the csv file name and second parameter specifies format parameter
- "csv.writer" is the inbuilt function used to write csv file with two parameters. First parameter specifies the file name to write and second parameter specifies dialect to be in excel format
- "writer.writerow" specifies the values to be written in row format to the csvfile

7.2 Feature Selection and Extraction

Feature extraction is a common step in many machine learning problems. It is a process that characterizes and reduces the dimension of raw input data. Features usually need to be selected carefully in order to obtain good classification performance. In this study, no specific feature extraction process is employed. The input data to the unsupervised learning algorithm are just the raw EMG signals. This is obtained by concatenating observed data values in to an input vector. For a 6 second time interval and 50 Hz sampling frequency, total number of samples from 8 EMG sensors will be 2400.

7.3 Normalization of dataset

When it comes to handling noisy data and missing values, K-means capability decreases. Data preprocessing techniques becomes effective applied to the datasets to make them cleaner, consistent and noise free. Hence Normalization enables us to eliminate the redundant and noisy data to ensure we are generating good quality clusters that increases that efficiency of the clustering algorithm. Thus, normalization turns out be an essential step before clustering as Euclidean distance is very sensitive to the changes in the differences.

The following plots in Fig 17 and 18 gives the normalized raw EMG values. In our work we use 'l2' normalization as this parameter for normalizer function in python

helps us to calculate the euclidean distance which is equivalent to sum of squared technique.

Fig 17 shows the normalized 8 channel EMG values for Gesture 1. Fig 18 shows the normalized single channel EMG value for Gesture 1.

$$d = |\mathbf{x} - \mathbf{y}| = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2} \,. \tag{7.1}$$

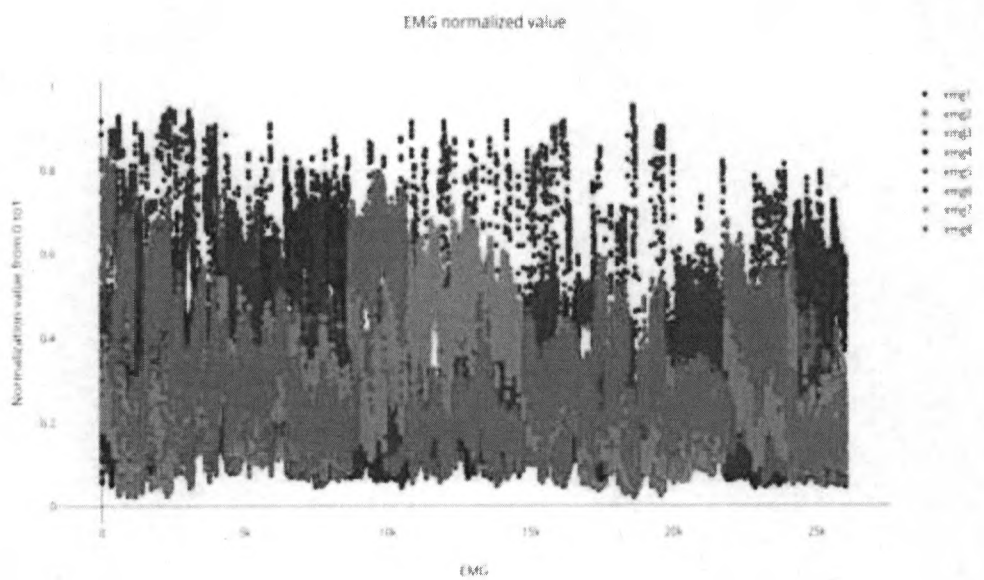

Fig 17. Plot of normalized data for raw value of EMG

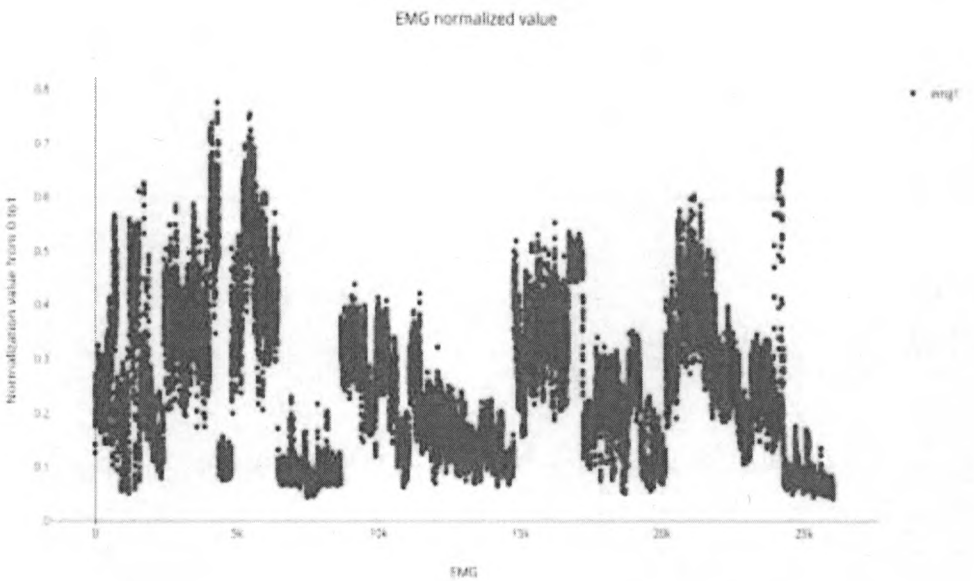

Fig18. Plot of normalized data for raw value of EMG

7.4 Data Shuffling

Shuffling, Training and Testing data: The entire normalized data set is shuffled to train and test data. In our work 70% of the data set forms the training set and 30% of the data set forms the test data. random_state is the seed used by the random number generator and in our case randomn_state is assigned as 123, it could be any random number. Our data set is in the form of pandas dataframe, so we use array parameter to pass the data set to randomly split our data set into train and test data. "test_size" parameter is passed to specify the portion of randomly shuffled data set to pick the test data set for model evaluation. In our case, test size is 30% of the entire data set. We

specify the parameter "train_size=None" to automatically allocate rest 70% of the data set to be training data set.

```
clus_train, clus_test = train_test_split(data_l2, test_size=0.3, random_state=123)
```

Fig19. Code Snippet for Training and Testing data split

7.5 ML Model training

```
#k-means clusters for range 1 to 9
clusters = range (1,9)
meandist=[]
scores=[]
for k in clusters:
        model=KMeans(n_clusters=k)
        cluster_label=model.fit(clus_train)
        clusassign=model.predict(clus_train)
        meandist.append(sum(np.min(cdist(clus_train,model.cluster_centers_,
'euclidean'), axis=1)) / clus_train.shape[0])
```

Fig20. Code Snippet for K Means model training

For unlabeled data, Clustering is used to find structure. For the dataset we don't know anything about, a clustering algorithm will discover groups of objects where the

average distances between the members of each cluster are closer than to members in other clusters.

First we need to determine the K value. We specify a range of values 1 to 9 to determine the K value. The data set is trained using k-Means clustering. As discussed in Section 4.1 and 4.2, the cluster range starts from 1 till a range of 9 assigned to variable "clusters". Mean distance is the feature extractor that calculates the Euclidean distance between the data points from the training data set.

1. In the second step new centroids are created by obtaining mean value of all the samples assigned to each previous centroid

2. In the third step, the algorithm keeps looping between computing old and new centroids until a threshold value where the centroids do not move significantly.

As seen in the Elbow method, the cut off value for clusters is seen at a value of 3. The clustering separation remains saturated beyond a centroid value of 3. Hence we assign cluster value as 3 to reiterate KMeans. Chapter 4 gives a detailed explanation of K Means computing.

Chapter 8

Result analysis

The main aim of the project was to demonstrate the overall correlation between time and EMG values implying that different clusters are formed for different gestures. While this result could be modeled using different regression analysis, the relationship between the data might not be linear. Hence K Means could serve as a valuable tool. K Means algorithm used for partitioning a set of data into k clusters and the result of such a partitioning technique is a list of clusters with our EMG values, which is not as visually appealing as the hierarchical methods. Silhouette analysis is considered that the graphical display will contribute to the interpretation of cluster analysis results.

Each cluster obtained from K Means will be represented by a silhouette. This is based on the comparison of its tightness and separation of the clusters. Silhouette displays which objects lie correctly within their cluster, and which other objects are merely somewhere in between clusters. The clustering will be displayed by merging the silhouettes into one plot giving an appreciation of the quality of the clusters and an outline of the data configuration. The average silhouette width gives an evaluation of clustering validity and can be used to select suitable' number of clusters. Silhouette analysis will be able to address the cohesion of cluster and cluster density. This method serves as a metric evaluation of our clustering analysis in our study.

The silhouettes built are useful when the proximities are on a ratio scale i.e., in our case it is Euclidean distances and when we are in need of concise and clearly separated clusters. Silhouette definition uses average proximities as in the case of group average linkage, which is known to work best in a situation with spherical clusters.

We need two things to construct a Silhouette:

1) Cluster partitioning of data obtained from application of K Means
2) Proximities of the objects

These are obtained from our K Means application from Section 7.5 results.

Silhouette coefficient measures how good an observation is clustered and it approximates the average distance between clusters. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters. Silhouette coefficient is calculated as below:

For each observation i the silhouette width si is calculated as follows:

For each observation i , calculate the average dissimilarity ai between i and all other points of the cluster to which i belongs. For all other clusters C , to which i does not belong, calculate the average dissimilarity $d(i,C)$ of i to all observations of C. The smallest of these $d(i,C)$ is defined as $b_i = \min_C d(i,C)$. The value of bi can be seen as the dissimilarity between i and its "neighbor" cluster, i.e., the nearest one to which it does not

belong. Finally the silhouette width of the observation *i* is defined by the formula: $Si=(bi-ai)/max(ai,bi)$ [12].

```
scores = []
range_values = np.arange(2, 9)

# print data

for i in range_values:
        print 'running'
        # Train the model
        kmeans = KMeans(init='k-means++', n_clusters=i, n_init=10)
        print 'running kmeans done'
        kmeans.fit(data)
        print 'running data done'
        score = metrics.silhouette_score(data, kmeans.labels_,
                metric='euclidean', sample_size=len(data))

        print 'done'
        print "\nNumber of clusters =", i
        print "Silhouette score =", score

        scores.append(score)

print 'score ready and appended'

# Plot scores
plt.figure()
plt.bar(range_values, scores, width=0.6, color='k', align='center')
plt.title('Silhouette score vs number of clusters')

plt.show()

# Plot data
plt.figure()
plt.scatter(data[:,0], data[:,1], color='k', s=30, marker='o', facecolors='none')
x_min, x_max = min(data[:, 0]) - 1, max(data[:, 0]) + 1
y_min, y_max = min(data[:, 1]) - 1, max(data[:, 1]) + 1
plt.title('Input data')
```

```
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())

plt.show()
```

Fig21. Code Snippet for Silhouette score for K Means model

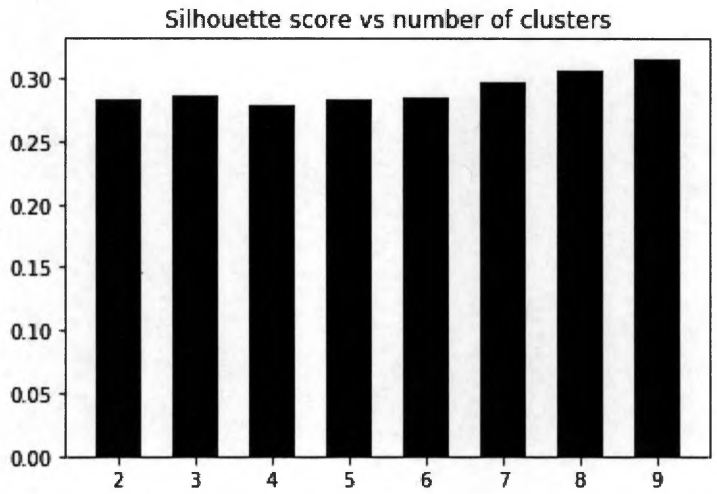

Fig22. Plot of Silhouette score vs number of clusters

Number of clusters = 2	Number of clusters = 3	Number of clusters = 4	Number of clusters = 5	Number of clusters = 6	Number of clusters = 7	Number of clusters = 8	Number of clusters = 9
Silhouette score = 0.284060557467	Silhouette score = 0.286289234787	Silhouette score = 0.278794533978	Silhouette score = 0.283005094784	Silhouette score = 0.284203363711	Silhouette score = 0.296507897002	Silhouette score = 0.306744215871	Silhouette score = 0.315849624191

Table 2: Silhouette score obtained in our study

Another revaluation method we use in our study is cross validation technique to determine the optimal number of clusters for K Means unsupervised clustering is Elbow Method. It is a cross-validation technique that verifies K Means clustering by Sum of squared errors method. From the silhouette analysis shown in Fig 22, it is observed that silhouette score remains highest for a cluster value of 9. This value is re-verified using Elbow method and as seen in Fig 24, the curve elbows at a value close to 5. The technique to determine K, the number of clusters, is called the elbow method. Elbow method cross - validates that any value above cluster value of 5, K Means performance remains high. The average distance between centroids and data points decreases from a value of 5 and it can be visually seen from Fig 24, for optimal number of cluster value of 8 the cluster coherence is highest that means the data points are tightly knit with the centroids.

```
Nc = range(1, 20)
kmeans = [KMeans(n_clusters=i) for i in Nc]
kmeans
score = [kmeans[i].fit(X_test).score(X_test) for i in range(len(kmeans))]
score
pl.plot(Nc,score)
pl.xlabel('Number of Clusters')
pl.ylabel('Score')
pl.title('Elbow Curve')
pl.show()
```

Fig23. Code Snippet of Elbow method for K Means model

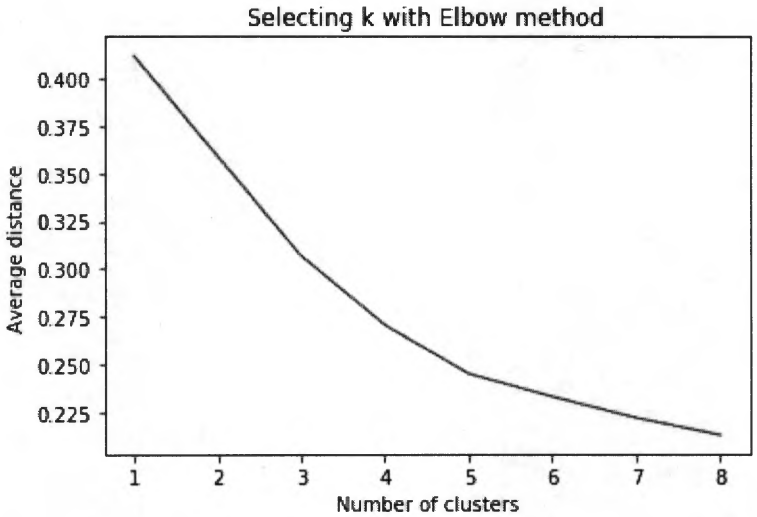

Fig24. Plot of Elbow method

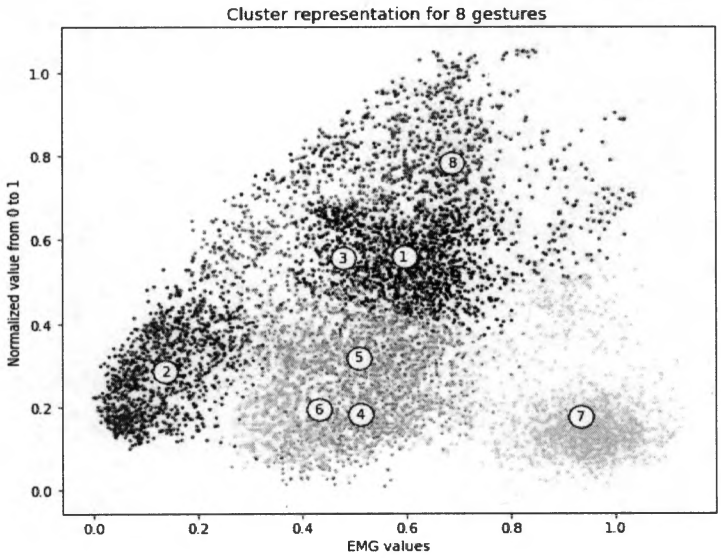

Fig25. Cluster plot of EMG

Cluster Number	1	2	3	4	5	6	7	8
No. of samples in cluster	4280 samples	3742 samples	4501 samples	2207 samples	1864 samples	4104 samples	2505 samples	2823 samples

Table 3: K Means clustering samples

Chapter 9

Conclusion and Future Work

EMG-based Gesture recognition has shown great promise for intuitive control of many human-machine interaction applications. Most EMG gesture recognition algorithms use supervised learning-based machine learning methods to interpret data and make decisions, which require appropriate labeling of all the training data. However, the performance of EMG gesture recognition can be affected by many factors such as environmental noises, motion artifacts, sensor shift, contact loss, and muscle fatigue. In order to maintain the gesture recognition performance, the system might need frequent re-training. Labeling the re-training data is not always easy especially for real-time, online gesture recognition systems. This project investigated the effectiveness of k-means clustering - an unsupervised learning technique, for hand gesture recognition based on unlabeled EMG data sets obtained from the Myo armband. The experiments for clustering five hand gestures based on three subjects' data did not show acceptable performance. This might due to several reasons. First, the defined gestures involved hand, wrist, and arm movements and might not be distinguishable by only EMG signals. In addition, the input data to the unsupervised learning algorithm were the raw EMG data without feature extraction. In the future work, we will try utilizing both EMG and IMU data, and applying windowing scheme and feature extraction methods to unsupervised learning-based gesture recognition.

References

[1] Malima, A., Ozgur, E. and Cetin, M. 2006. “A Fast Algorithm for Vision-Based Hand Gesture Recognition for Robot Control”. In Proc. of 14th IEEE Signal Processing and Communications Applications.

[2] Sachs, E. 1989. “3-Draw: a three dimensional computer aided design tool”. In Proc. of IEEE International Conference on Systems, Man and Cybernetics.

[3] Song, Yang, L. Goncalves, and P. Perona. "Unsupervised Learning of Human Motion." IEEE Trans. Pattern Anal. Machine Intell. IEEE Transactions on Pattern Analysis and Machine Intelligence 25.7 (2003): 814-27

[4] Trabelsi Dorra, Samer Mohammed, Yacine Amirat and Latifa Oukhellou, “Activity Recognition Using Body Mounted Sensors: An Unsupervised Learning Based Approach.” The 2012 International Joint Conference on Neural Networks (IJCNN), 2012.

[5] Nguyen, A., Darren Moore, and I. Mc.Cowan. "Unsupervised Clustering of Free-Living Human Activities Using Ambulatory Accelerometry." Proceedings of the 29th Annual International Conference of the IEEE EMBS Cité Internationale, Lyon, France, 23 Aug. 2007.

[6] Stephen O’Hara, Yui Man Lui and Bruce A. Draper, “Unsupervised Learning of Human Expressions, Gestures, and Actions”, Partially supported by DARPA Contract.

[7] Zhang Qiu-yu, Lu Jun-chi, Zhang Mo-yi, Duan Hong-xiang and Lv Lu, "Hand Gesture Segmentation Method Based on YCbCr Color Space and K-Means Clustering", In Proc.of International Journal of Signal Processing, Image Processing and Pattern Recognition , 2015 ,Vol. 8, No. 5, pp. 105-116 , ISSN: 2005-4254 IJSIP.

[8] Abhijit Kane, "Determining the number of clusters for a K-means clustering algorithm", In Proc. Of International Journal of Computer Science and Engineering, 2012, ISSN : 0976-5166 Vol. 3 No.5.

[9] Wang, Shouyi & Chaovalitwongse, Wanpracha & Babuska, Robert. (2012), " Machine Learning Algorithms in Bipedal Robot Control", Systems, Man, and Cybernetics, Part C: Applications and Reviews, in IEEE Transactions on. 42. 728-743. 10.1109/TSMCC.2012.2186565.

[10] "Myo Gesture Control Armband." Myo Gesture Control Armband | Wearable Technology by Thalmic Labs, www.myo.com/.

[11]Konrad, Peter. "The ABC of EMG", retrieved from hermanwallace.com/download/The_ABC_of_EMG_by_Peter_Konrad, Version 1.0 April 2005.

[12] Rousseeuw, Peter J. "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis." Journal of Computational and Applied Mathematics, vol. 20, 1987, pp. 53–65., doi:10.1016/0377-0427(87)90125-7.

[13] Arpita Ray Sarkar, G. Sanyal and S. Majumder. 2013. "Hand Gesture Recognition Systems: A Survey". In Proc of International Journal of Computer Applications.

[14] Yang, Z., Li, Y., Chen, W. and Zheng, Y. (2012) "Dynamic Hand Gesture Recognition Using Hidden Markov Models". In Proc. of 7th Int. Conf. on Comp. Sc. and Education.

[15] Oka, K., Sato, Y. and Koike, H. (2002) "Real-time Tracking of Multiple Fingertips and Gesture Recognition for Augmented Desk Interface Systems". 5th IEEE Int. Conf. on Automatic Face and Gesture Recognition.

[16] Kohn, B., Belbachir, A. N. and Nowakowska, A. 2012. "Real-time Gesture Recognition using bio inspired 3D Vision Sensor". In Proc. of IEEE Comp. Society Conference on Comp. Vision and Pattern Recognition Workshops.

[17] Kasprzak, W., Wilkowski, A. and Czapnik, K. "Hand Gesture Recognition based on Free-Form Contours and Probabilistic Inference". Int. J. of Applied Math. and Comp. Sc. 22(2) (2012), 437- 448.

[18] Elmezain, M., Al-Hamadi, A., Sadek, S. and Michaelis, B. 2010. "Robust Method for Hand Gesture Spotting and Recognition Using Hidden Markov Model and Conditional Random Fields". In Proc. of IEEE Int. Symposium on Signal Processing and Information Tech.

[19] Meena, S. 2011. “A Study on Hand Gesture Recognition Technique”. M. Tech. thesis. NIT, Rourkela (India).

[20] Krause, Andreas, Asim Smailagic, Daniel P. Siewiorek, and Jonny Farringdon. "Unsupervised, Dynamic Identification of Physiological and Activity Context in Wearable Computing." Unsupervised, Dynamic Identification of Physiological and Activity Context in Wearable Computing (2003).

[21] LaViola, Joseph J. “A Survey of Hand Posture and Gesture Recognition Techniques and Technology.” ACM Digital Library, Brown University, June 1999.

[22] Lara, Oscar D., and Miguel A. Labrador. “A Survey on Human Activity Recognition Using Wearable Sensors.” IEEE Communications Surveys & Tutorials, vol. 15, no. 3, 2013, pp. 1192–1209., doi:10.1109/surv.2012.110112.0019

[23] “Scikit-Learn: Machine Learning in Python - Scikit-Learn 0.16.1 Documentation.” Scikit-Learn: Machine Learning in Python - Scikit-Learn 0.16.1 Documentation, scikit-learn.org/.

[24] “Modern Visualization for the Data Era.” Plotly, plot.ly/.