PERSON IDENTIFICATION AND ANOMALY DETECTION USING GAIT PARAMETERS EXTRACTED FROM TIME SERIES DATA

by

SUHAS MANDIKAL RAMA KRISHNA REDDY

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2017

To my parents.

ACKNOWLEDGEMENTS

I would like to thank Dr. Manfred Huber for giving me the opportunity to research on gait analysis. I appreciate his efforts in taking time to advice whenever being approached. I would like to thank Dr. Farhad A Kamangar and Dr. Gergely V Zaruba for taking time to serve on my committee and for their valuable comments. I would also like to thank Oluwatosin Oluwadare for taking time to explain to me his research.

I would like to thank Irie Bito for his constant support and motivation. I would also like to thank my friend Sonali Marne for formatting this report according to the requirements. Lastly, I would like to thank my family for believing in me and supporting me through this journey.

April 24, 2017

ABSTRACT

PERSON IDENTIFICATION AND ANOMALY DETECTION USING GAIT PARAMETERS EXTRACTED FROM TIME SERIES DATA

SUHAS MANDIKAL RAMA KRISHNA REDDY, MS

The University of Texas at Arlington, 2017

Supervising Professor: Manfred Huber

Gait generally refers to the style of walk and is influenced by a number of parameters and conditions. In particular, chronic and temporary health conditions often influence gait patterns. As such conditions increase with age, changes in gait pattern and gait disorders become more common. Changes in the walking pattern in the elderly can suggest neurological problems or age related problems that influence the walk. For example, individuals with parkinsonian and vascular dementias generally display gait disorders. Similarly, short term changes in muscle tone, strength, and overall condition can reflect in gait parameters. Analysis of the gait for abnormal walk can thus serve as a predictor for such neurological disorders or disorders related to age and potentially be used as a means for early detection of the onset of chronic conditions or to help prevent falls in the elderly. In our research we try to build personalized models for individual gait patterns as well as a framework for anomaly detection in order to distinguish individuals based solely on gait parameters and in order to try to detect deviations in walking based on these parameters.

In this thesis we use time series data from pressure monitoring floor sensors to real-time segment walking data and separate it from data representing other activities like standing and turning by using unsupervised and supervised learning. We extract spatio-temporal gait parameters from relevant walking segments. We then model walking of individuals based on these parameters to predict deviation in walking pattern using the Support Vector Data Descriptor (SVDD) method and the One Class Support Vector Machine (OCSVM) for anomaly detection. We apply these models to real walking data from 30 individuals to attempt person identification to demonstrate the feasibility of building personalized models.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

Table Page

LIST OF ALGORITHMS

Chapter 1

INTRODUCTION

“Gait is the manner or the style in which people walk”. Walking is a complicated process that involves the brain, spinal cord, nerves, muscles, bones and joints. Gait analysis serves to assist in the treatment of individual persons and to improve the understanding of gait through research. [15].

1.1 Basic Physiology of Movement

Gait starts with activity in several areas of the brain. Nervous impulses are sent down the spinal cord to activate the motor nerves to allow muscle contraction. Feedback from sensors from the muscle and the surrounding areas to the brain causes modification in walking pattern. The areas in the brain that are affected during walk are the basal ganglia which is responsible for learning and walking patterns, the motor cortex which interacts with the muscles to help in movements, the cerebellum that provides the feedback on the correctness of the activity, and the spinal cord and the brainstem motor nuclei which play a role in generating and controlling movements.[15].

1.2 Conditions Benefiting from Gait Assessment

Major diseases that affect the neuromuscular and musculoskeletal systems that lead to disorders in gait include cerebral palsy, parkinsonism, muscular dystrophy, osteoarthritis, stroke, spinal cord injury, myelodysplasia and multiple sclerosis. [15] Cerebral palsy is caused by damage to the brain.

1.3 Motivations for Gait Analysis

Cerebral palsy is caused by damage to any of the parts of the brain concerned with walking. Cerebral palsy caused due to the degeneration of the basal ganglia, causes shuffling gait [16]. Parkinson's disease causes reduction in step length and speed [15].

Gait problems become increasing with age. Neurological diseases, arthritis and acquired foot deformities are common causes of abnormalities in gait in the elderly. Neurological diseases like neuropathies, stroke, and parkinsonian syndrome are present in 30 -50 % of elderly individuals who are diagnosed for abnormal gait [19][20][21]. Non Alzheimer's dementia such as vascular and parkinsonian dementia show symptoms of gait disorder [19][22][23]. Non-Alzheimer's dementia and vascular dementia in particular could be strongly predicted based on neurologic gait abnormalities [19].

1.4 Our Approach

Neurological problems, musculoskeletal problems and temporary conditions such as injury, neck pain and back pain causes change in gait patterns. Abnormal gait, in turn, can cause falls in elderly. Falls are one of the major causes of hospitalization and early death in the elderly. In order to predict neurological and musculoskeletal problems at the onset of the disease before it becomes chronic and to predict falls, we propose an approach to predict change in walking pattern by continuously monitoring gait using pressure monitoring floor sensors. We start with performing activity recognition to differentiate relevant walking segments for gait parameter extraction from standing, short feet movement and turning segments. We extract spatio-temporal parameters from the strides extracted. We then use spatio-temporal gait parameters to build individual models for person identification and anomaly detection. We use an SVM with linear, polynomial and RBF kernels along with One Class SVM (OC-SVM) and Support Vector Data Descriptor (SVDD) methods to perform person Identification. We also use OC-SVM and SVDD to identify anomaly in walking.

1.5 Existing Approaches

Existing approaches for gait analysis generally make use of devices worn on the body, video surveillance cameras, electrodes mounted on the skin, needles pierced into the muscle, or kinematic systems to name a few, to obtain data for detection of abnormality in gait. Traditionally, clinical analysis is carried out to determine neurological or musculoskeletal disorders using the measurements obtained from these approaches.

Modern day approaches involve collection of data using these devices and building models to predict deviation in walking pattern. Most of these approaches use devices worn on the body or inserted into the shoe or video surveillance cameras to extract parameters for model building. In practice, however, it may not be always feasible to wear such devices to obtain unobtrusive data for analysis. It would also feel uncomfortable to be monitored by cameras. These devices may actually hinder a person from normally exhibiting their regular walking behavior. We propose an approach where we do not use devices worn on the body and where we are not being recorded by any cameras. In our approach we use pressure monitoring sensors embedded beneath the floor to gives us continuous data for detection of deviation in normal walking pattern. There we largely avoid the white coat effect while obtaining measurements that reflect the actual behavior.

Chapter 2

METHODS FOR GAIT ANALYSIS

Here we try to briefly describe the different existing approaches for gait analysis. The approaches listed here are not exhaustive but aim to cover most of the popular approaches. Gait analysis can be done with and without having technological aids. Simple techniques can be used to manage clinical problems. [15].

2.1 Visual Gait Analysis

The basic form of Visual Gait Analysis is observation made through the human eye. Though visual gait analysis can be used to determine serious gait abnormalities, it has serious limitations. It cannot be used to observe high speed events, only movements can be observed and not forces, and it depends highly on observation skills. [15]. Moreover, objects in the environment, such as furniture, as well as wide and loose clothing, including night gowns, can severely limit the data that can be extracted using visual observations and computer vision techniques.

2.2 Examination by Video Recording

Difficulty of observing high speed events by the naked eye can be overcome by video recording. Since a permanent record is available, the subject is required to do only a few walks. Video recording however does not provide quantitative information which could be very useful for analysis. Though quantitative information can be derived from the video it can turn out to be inaccurate because of the viewing angle and the distortions from the recording camera. [15].

2.3 Analysis using Gait Parameters

Robinson & Smidt in their research on quantitative gait evaluation [17] provide objective information regarding the patient’s gait. They focus on extraction of temporal and distance factors such as stride length, step length, cadence and velocity using some basic equipment such as stopwatch, portable tape recorder and measuring tape. [15].

Footswitches and Instrumented Walkways for timing of gait cycle

Footswitches placed beneath the heel and the fore foot of the shoe are used to measure the timing of initial contact, heel strike and toe off. Gait cycle time and swing phase duration can be calculated from the obtained data. Double Support and single support times are measured by placing switches underneath both feet. [15].

Instrumented walkways are used to obtain timings of foot contact and position of foot by making use of a conductive walkway and electrical contacts on the shoes to complete the circuit to provide the same information as switches. [15].

In our research we make use of such temporal and distance factors to model the walking of a subject but instead of mounting sensors to the foot of the individual we use sensors embedded in the home environment to obtain this data unobtrusively..

2.4 Electrogoniometers

Electrogoniometers are devices used to make continuous measurements of the angle of a joint. Interactions between the different joints can be analyzed through a cyclogram to identify characteristic patterns. Potentiometer devices and flexible strain gauges are devices that need to be worn on the body get the measurements. The potentiometer measures the joints angle in degrees of the knee and the hip. Flexible strain gauges are strips that are attached to the joints to measure the angles. [15].

Usually these devices take significant time and precision to be mounted on the body and are thus only practical in clinical settings for short term analysis.

2.5 Pressure On the Floor

Pressure sensors can be used to identify disease like diabetic neuropathy and rheumatoid arthritis where the pressure on the floor is excessive. The pressure monitoring devices may be floor mounted or in the insole within a shoe. Glass plate examination using a mirror can also be carried out to see which parts of the foot are in contact with the floor when the person is walking and to obtain an idea of the pressure being applied by the part of the foot. Direct pressure mapping systems transfer ink to paper, producing greater imprints in areas of higher pressure. Other approaches include use of aluminum foil and carbon paper. Force sensor systems are also used to measure the vertical forces beneath the area of the foot. There are in shoe devices that are used to measure the pressure and give clinically useful results. [15].

2.6 Electromyography

The measurement of the electrical activity of the contracting muscle is called Electromyography (EMG). Surface electrodes attached are used to measure the electrical activity. Another approach is by using fine wire electrodes using hypodermic needles pierced into the muscle to record electric activity. [15].

2.7 Energy Consumption

Other approaches involve ways to measure energy expenditure while performing an activity. The ratio of oxygen to carbon dioxide in the exhaled air gives the measure of the oxygen consumed [15]. Heart rate monitoring is often used as a substitute for oxygen consumption due to a close relationship between oxygen consumption and heart rate [18].

2.8 Accelerometers

Accelerometers are used to measure the acceleration to analyze gait parameters. Heel strike transients, which are high acceleration events, are measured using accelerometers.

2.9 Force Platforms

Force platforms measure the ground reaction forces as the subject walks across the platform. The output of the force platform gives the acceleration of the body in three dimensional space [15].

2.10 Kinematics

The geometric description of motion is referred to as kinematics. The motion is measured in terms of displacements, velocities and accelerations. Kinematic measurements can be made in two dimensions or three dimensions [15].

Chapter 3

RELATED WORK

3.1 Person Identification Based on Gait

There is a significant amount of research on person identification based on gait. Most of the approaches for person identification are based on video data.

C. BenAbdelkader et al [24] perform person identification from spatio-temporal features extracted from a video. The features they extracted from the walk were estimated stride length and cadence. With data of 17 individuals the person was verified with an error rate of 11% and correctly identified with a probability of 40%.

Liang Wang et al [25] propose a technique for person identification based on spatial-temporal silhouette analysis by background subtraction on a video. Principal component analysis was applied to reduce the dimensionality of time-varying distance signals. Supervised learning was performed on a lower dimensional Eigenspace to recognize individuals.

C. BenAbdelkader, R. et al [26] extended their previous work [24] of person identification based on stride and cadence by adding height as an additional discriminant. They claim that results significantly improved by adding height. Testing over 45 individuals with height and stride parameters they obtain a detection rate of 49%. With stride parameters alone, they obtain an accuracy of 21 %. They also claim that the approach is good for low resolution images.

Zhang, Z., & Troje, N. F. [27] put forward a technique to identify individuals independent of viewpoint using three-dimensional linear models and Bayes rule.

Geiger, J. T et al [28] present a system to identify individuals based on acoustic gait recognition. The sounds are characterized by gait patterns influenced by movement of arms and legs. Cepstral features are extracted from the recorded audio signals. Hidden Markov models are used for classification. An identification rate of 65.5 % was obtained in classifying 155 individuals.

3.2 Anomaly Detection in Gait

Detection of anomaly in gait early on can prevent neurological diseases or age related disorders from becoming chronic at the onset of the disease. Early detection of change in walking pattern can also prevent falls in the elderly. There are numerous approaches proposed for anomaly detection in gait. The thesis “Gait analysis on smart floor” [10] summarizes the approaches. Some of the recent approaches are described here:

Cola et al [32], propose a method to detect deviation in gait by using a wearable accelerometer. A single waist mounted accelerometer is used to continuously monitor gait in an unsupervised fashion. The experiment was carried out on 30 subjects who simulated knee flexion impairment. They claim to have obtained an accuracy of 84% in recognition of abnormal gait segments of length 5 seconds. The knee is tied with a strap on one of the limbs, two straps are tied around the knee of the same limb and each knee is tied around with a strap to simulate mild, severe and a case of both knees having impairment. Nearest neighbor technique is used for prediction of abnormality in walk.

Begg et al [33] propose a method to automatically classifiy gait patterns of young and the old. Gait data is extracted from individual Minimum Foot Clearance histogram-plots and Poincaré-plot images obtained from continuously walking data of 20 minutes on a treadmill. Cross validation using SVM resulted in 83.3% accuracy in comparison with 75% using neural network.

Martinez et al [34] predict fall risk in elderly from collecting gait data from a smartphone. The data is collected in parallel from a pressure sensitive walkway and a smartphone. Data collected from the pressure sensitive walkway is used as a ground truth. They make use of one class SVM (OC-SVM) [7] and support vector data descriptor SVDD [6] methods to detect anomalies. Inertial data collected from the phone is used for anomaly detection. Falls risk ratio of each individual was calculated using walkway measurements. Based on falls risk ratio 14 participants were determined as at high risk from falling and 10 as not. Results indicate OC-SVM with RBF kernels to be better.

Nguyen et al [35] propose skeletal based abnormal gait detection by making use of normal gait model. The model is based on human joint positions in time series. The feature vector contains values that explain the relationship between each pair of bone and joints located in the lower body. Clustering is used to form code words using those feature vectors. Sequences of code words of normal human gait cycles are used for training the Hidden Markov Model. If the likelihood of belonging to the normal data is below a threshold, the data sequence is flagged as anomalous.

3.3 Anomaly Detection Using SVDD And OCSVM

One Class Support Vector Machines (OCSVM) are classifiers used for novelty detection or outlier detection. These classifiers are used when little information or no information about the outlier data is known. The classifiers can be trained only on data belonging to the target class i.e. where the data does not contain outlier data. Variants of the classifier can be trained on target data and outlier data as well. Some of the existing work for anomaly detection using one class Support Vector Machines are as detailed below.

Zhang et al [28] propose a method for fall detection using one class SVM on data from wearable sensors. They use the data coming from a wearable tri-axial accelerometer to capture the movement of the subject. The target data is falls of younger individuals and the outlier data is data from regular activities from younger and older individuals. Data was collected from low risk fall down, where the subjects fell on support cushion, high risk fall down when they fall down on slope and stairs, along with low and high intensity activities. They use a Support Vector Data Descriptor to build models. Using the SVDD they were able to classify the data of fall and daily movements.

Larry M. Manevitz and Malik Yousef [29] implement variants of one class SVM for document classification. They make use of Schölkopf et al ‘s [7] one class SVM and outlier SVM [6] and other approaches like Neural Networks. Schölkopf et al ‘s one class SVM does better on smaller data sets while outlier SVM and Neural Networks perform better on larger data sets.

Zhou et al [30] use one class Support Vector Machine to perform image segmentation to extract brain tumors from magnetic resonance images. The approach segments brain tumor images with high accuracy.

Zaho Y et al [31] use one class support vector machine for customer churn prediction. They used linear, polynomial and Gaussian kernel to perform the classification. The Gaussian kernel gave the better results. In comparison with artificial neural network, decision tree, Naive Bayes and SVM with Gaussian kernel classifiers, the SVM with Gaussian kernel outperformed all approaches.

Chapter 4

TECHNICAL BACKGROUND

4.1 Gait Terminologies and Parameters

When walking, one limb acts as a support while the other limb moves forward to a new support site. Reversing the roles and repeating the process, the individual reaches the destination. People usually make contact with the floor on a heel strike; other individuals land their foot flat, so the contact of the foot with the floor can be termed as initial contact or heel strike. [1]

In a gait cycle there are two phases, the swing phase and the stance phase. The entire period beginning with the initial contact during which the foot is on the ground is called stance phase. The swing phase starts with the toe off and is the period during which the limb is in the air until the initial contact occurs. The stance and swing action of one limb is called a stride. It is also called a gait cycle. A stride consists of two steps. The step length is the distance between the heel strikes of the limbs. [1].

During single support, one of the limbs supports the body while the other limb is in the air. When both the limbs support the body it is called double support. The start of double support is when the leading limb which was in the air touches the ground while the trailing limb is still touching the ground. During running, double support is not present. [1]. Figure 3-1 shows the swing and the stance phase during gait. The swing and the stance phase make up the gait cycle.

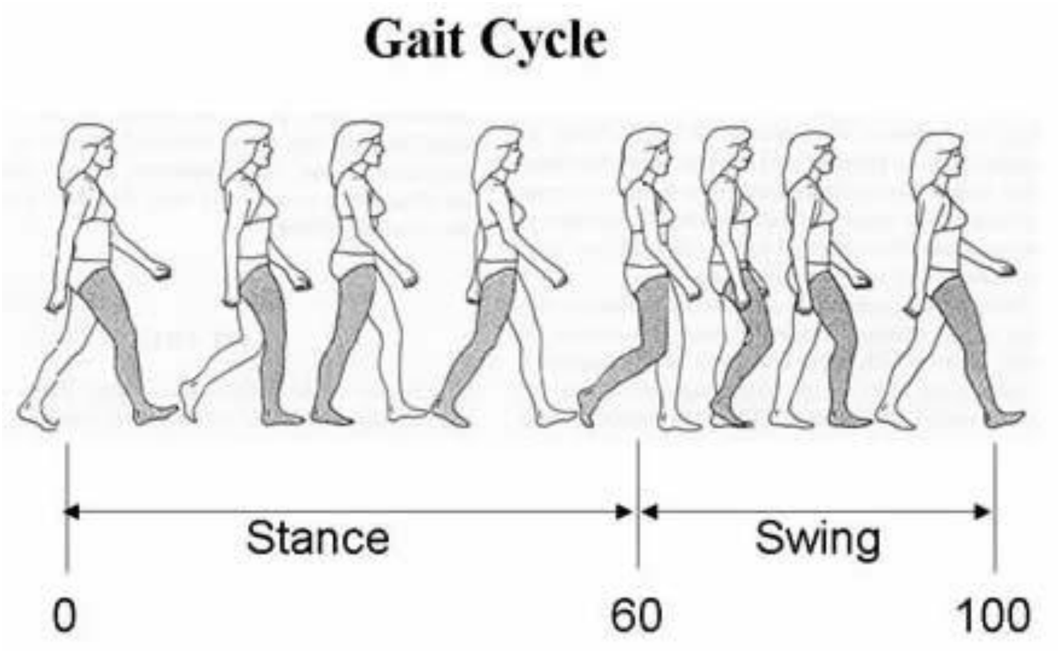


Figure 4-1 Swing phase and stance phase [14]

4.1.1 Gait Parameters

Gaits are frequently characterized in terms of a number of gait parameters capturing common properties of the individual’s gait.

Stride Length

Stride length is the distance from the heel of the first limb to the heel of the same limb. Two steps equal a stride.

Step Length

This is the distance from the heel of one limb to the heel of the other limb.

Single Support Time

Single support time is the time it takes for the initial contact or the heel strike and the toe off of the same foot.

Step Frequency

Step frequency is the number of steps the subject walked during a time interval.

Step Ratio

Step ratio is the ratio between the two step lengths. Two steps are made during a stride. In other words it is the ratio of the distance between two steps of the stride.

Average Speed

Average speed is the distance covered between the first heel strike and the third heel strike divided by the time it takes to cover the distance.

Average Step Length

It is the average of the two step lengths of the stride.

4.2 Unsupervised Learning

In unsupervised learning the training data does not contain any output information. The input is a set of training examples $x_1, x_2, x_3 \ldots x_N$. The task of unsupervised learning is to spontaneously find patterns and structure in input data. Unsupervised learning can be viewed as a method to create a higher level representation of data. Clustering is unsupervised learning. [4].

Clustering

Partitioning the data set into subsets is called clustering or cluster analysis. The result of clustering produces objects in a cluster that are more similar within a cluster and objects between clusters being more dissimilar. Using the same data set as input produces different clustering for different clustering algorithms. Previously unknown information can be discovered using clustering. Insight into data distribution can be obtained using cluster analysis. Clustering can be used as a preprocessing technique to identify clusters so that they can be used for classification. [2]

In our research we use clustering to group similar data segments. The data segments belong to different activities.

Similarity Measure

The similarity between two points can be determined by the distance between them in Euclidean space. There are other approaches to find similarity between two points. [2]

4.3 Clustering Methods

Major clustering methods are described below.

4.3.1 Partitioning Methods

Given n points, the data is divided into k groups. The partitioning is based on distance. Objects in the same cluster are more similar and objects between clusters are more dissimilar. Based on this criterion, objects are clustered. To find global optimality it would require exhaustive enumeration of all possible partitions. To prevent the exhaustive enumeration, approaches such as k-means and k-medoids are used to approach global optimum. [2].

4.3.2 Density-based methods

Density-based methods cluster objects based on the distance between objects. The idea is to grow clusters as long as their density (the number of objects) in the neighborhood exceeds some threshold. Density based clustering can be used to filter out noise or outliers. [2].

4.3.3 Grid-based methods

The object space is quantized into a fine number of cells that form the grid structure. The clustering operations are performed on the quantized space. The approach is faster as it does not depend on the number of data points but rather on the number of cells. Grids are used in spatial data mining problems including clustering. Grid based methods can be integrated with density based methods and hierarchical methods. [2].

4.3.4 Hierarchical Methods

Hierarchical methods Build a hierarchy of clusters using data objects. Hierarchical methods can be of two types, agglomerative or divisive. The agglomerative hierarchical clustering is a bottom up approach where each initial cluster is a singleton cluster, and the singleton clusters are merged to eventually form a single cluster which includes all the points. Divisive clustering is an approach where all the points belong to one cluster initially and the cluster is divided into groups, eventually resulting in a single point in each cluster. The formed clustering is cut to obtain clusters [2].

In our research we use hierarchical clustering over other approaches as we do not use Euclidean distance for similarity measure, but instead use spectral coherence to compare similarity between frequency spectra. When using k-means algorithm, the number of clusters k must be decided ahead of time. K- means clustering also uses Euclidean distance for clustering. Density based approaches can only find spherical shaped clusters and discovering clusters of arbitrary shapes becomes difficult. Grid based methods are more suited to spatial data mining problems. For these reasons we use Hierarchical Clustering. [2].

4.4 Linkages for hierarchical clustering

Agglomerative or divisive clustering uses a distance measure to find the distance between two clusters, where each cluster is a set of objects. Some of the popular linkage measures are:

- Single Linkage
- Complete Linkage
- Mean Method
- Average Linkage

4.4.1 Single Linkage

When using the single linkage criterion we use the smallest distance between the two clusters. The smallest distance is the distance between the two most similar points belonging to two different clusters [2]. Figure 4-1 is an example for single linkage.

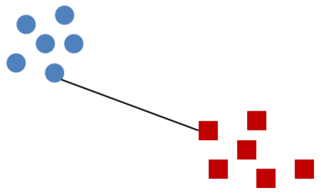


Figure4-2 Single Linkage

4.4.2 Complete Linkage

When using the complete linkage criterion we use the maximum distance between the two clusters. The maximum distance is the largest distance between the two points belonging to two different clusters [2]. Figure 4-2 is an example for complete linkage.

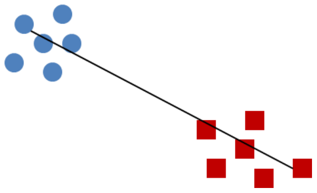


Figure 4-3 Complete Linkage

4.4.3 Mean Method

Mean linkage uses the distance between the centroid of one cluster to the centroid of another cluster. The centroid is computed by averaging the vectors of each point belonging to the cluster. [2].

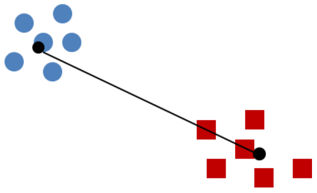


Figure 4-4 Centroid Linkage

4.4.4 Average Linkage

Average linkage uses the average of distance between all points between the two clusters. Clusters are merged at each stage based on the smallest distance between the two clusters. [2]

In our research we use the complete linkage criterion to merge the clusters. Following is an example of 5 points being merged using complete linkage. The matrix is a similarity matrix. The values of the matrix are similarity values between any two points. Similarity and distance are inversely proportional [2]. A value of 1 represents the highest similarity and a value of 0 represents the lowest similarity. Clusters at each level are merged based on the highest similarity. When two clusters are merged, the new similarity score for the merged cluster is the lowest similarity score (highest distance) of the two clusters.

Appendix A shows the working of the hierarchical clustering algorithm using complete linkage and a comparison between single linkage and complete linkage.

4.5 Supervised Learning

Given a training set containing input-output pairs $(x_1, y_1), (x_2, y_2), \dots \dots .(x_N, y_N)$, where x_i is the input and y_i is the corresponding output, the task of supervised learning is to discover a function *h* approximating the true function *f*, where y_i is generated by an unknown function y = f(x). [3]. Here *h* is the hypothesis that performs well on the unseen samples. The hypothesis *h* is obtained by learning, which is searching the space of hypotheses to find the one that best approximates f. Classification is supervised learning. [3].

Classification

Given the training data with class labels, the objective is to find a function or a model that that distinguishes the classes [2].

In our research, we us classification to differentiate activities like standing from walking, identifying individuals based on their gait parameters and to detect anomalies in the walking pattern.

4.5.1 Decision Tree

A decision tree is a classifier learned from class labeled training tuples. It is a flowchart like structure where each node denotes a test on a splitting attribute. The leaf nodes hold the class labels. The splitting nodes are represented by rectangles and the leaf nodes are represented by circles. Decision trees can have binary splits and non-binary splits. When the new data point is tested against the decision tree, it follows a branch that eventually leads to the name of the class. The class label is assigned to a test point. Figure 4-4 shows a decision tree for buying a computer. [2].

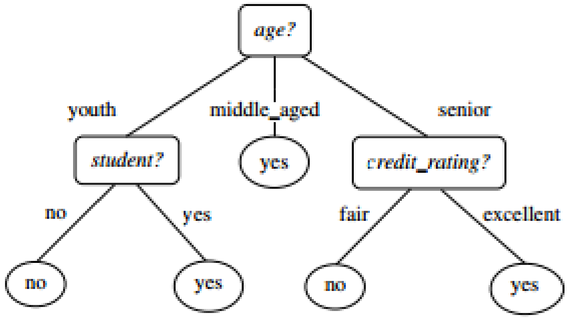


Figure 4-5 An example decision tree for buying a computer. [2]

Attribute Selection Measure

There are three popular attribute selection measures – Information Gain, Gain Ratio and Gini Index by which an attribute can be split. Each of these attribute selection measures outputs a numeric value associated with an attribute. This numeric value is used in deciding the attribute to split by. [2].

Gini Index

Here we describe constructing decision trees using Gini. Gini Index is used in Classification and Regression Trees (CART). The CART algorithm is used to implement decision tree in our research. The CART algorithm can be used for classification and for regression. In regression the output is a real value. Gini Index measures the impurity of D, a data partition or a set of tuples, as

$$Gini(D) = 1 - \sum_{i=1}^{m} {p_i}^2$$

where p_i is the probability the point in the training data set belongs to class C_i and m is the number of classes. p_i is estimated as $\frac{|C_{i,D}|}{|D|}$. [2].

The CART algorithm performs binary split. When performing a binary split, a weighted sum of impurity of each resulting partition is computed. [2].

$$Gini_A(D) = \frac{|D1|}{|D|} Gini(D_1) + \frac{|D2|}{|D|} Gini(D_2)$$

A reduction in the impurity by the binary split is given by

$$\Delta Gini(A) = Gini(D) - Gini_{(A)}(D)$$

The attribute that gives the highest reduction in the impurity is chosen as the splitting attribute. This technique holds for both, discrete and continuous valued attribute.

Continuous valued attributes are made categorical by using bins. Each possible split point is considered for continuous valued attributes. [2].

Appendix A shows the working of CART algorithm.

4.5.2 Gaussian Naïve Bayes Classifier

Another classifier used in this research is the Gaussian Naïve Bayes Classifier. The Gaussian Naïve Bayes classifier is used for continuous valued inputs.

$P(Class|Features)$ can be found by computing:

$$\frac{P(Features|Class)P(Class)}{\sum_i P(Features|Class_i)}$$

Ignoring the normalizing factor we compute $P(Features|Class)P(Class)$ where $P(Class)$ is the prior probability. [2].

A Gaussian has two parameters, mean and variance. The $p(f_i|Class)$ can be computed by plugging in the value of the feature f_i, using the mean and the variance of that feature.

$$\frac{1}{\sqrt{2\pi\sigma^2}}exp^{-\frac{1}{2}(\frac{|f_i-m|^2}{\sigma^2})}$$

Here, f_i is the input feature, m is the mean of the feature and σ is the standard deviation of the feature. $Max_{class_i}(P(Features|Class_i)P(Class_i))$ gives the class to which the set of features are more likely to belong. [2].

4.5.3 Support Vector Machines

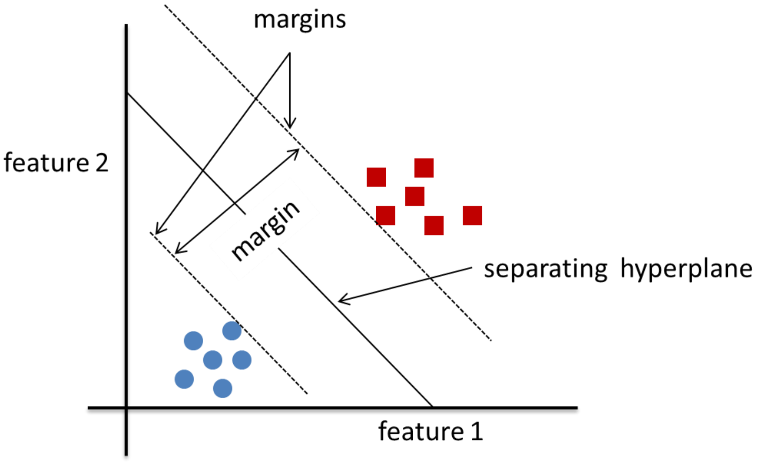


Figure 4-6 SVM on linearly separable data [2].

4.5.3.1 SVM for Linearly Separable Data

Another classifier used in our research is the Support Vector Machine (SVM). Variants of SVM are used for classification purposes. Let D be the data set given as (X_i,$\boldsymbol{y_i}$), where X_i is the training tuples associated with class labels y_i. y_i takes a value +1 or -1 indicating the class. There are infinitely many hyperplanes that can separate the two classes. The hyperplane is also called the decision boundary. The SVM finds the maximum marginal hyperplane. The margins are parallel to the hyperplane and lie on the support vectors. Figure 3-9 shows the case of a linear SVM. [2].

The separating hyperplane is written as:

$$W.X + b = 0,$$

where W is the weight vector and b is the bias. The length of the weight vector is equal to the number of features. $W.X + b > 0$ is satisfied for the points lying above the separating hyperplane. $W.X + b < 0$ is satisfied for the points lying below the separating hyperplane.[2].

For y_i = 1, the data points satisfy the equation

$$WX + b \ \geq 1 \qquad\qquad (1)$$

where W is the weight vector, X is the input vector and b is the bias. This equation holds because all of the points belonging to y_i = 1 are on one side of the margin.

Similarly, for y_i = -1, it satisfies the equation

$$WX + b \ \leq -1 \qquad\qquad (2)$$

This equation holds because all of the points belonging to y_i = -1 are on one side of the margin. [2].

Combining the two equations results in:

$$y_i(WX + b) \geq 1, \forall_i \qquad\qquad (3)$$

Weights are minimized to maximize the margin. The width of the margin is given by $\frac{2}{||W||}$ where $\left||W|\right|$ is the Euclidean norm of W, which is equal to$\sqrt{W.W}^2$. [2].

Rewriting Equation 3 using Lagrangian formulation and solving for the solution using Karush – Kuhn – Tucker conditions a separating hyperplane is obtained. The separating hyperplane is shown below. The equation makes use of Largange multipliers α_i.

$$\sum_{i=1}^{l} y_i \alpha_i X_i X^T + \ b_0$$

Here y_i is the class label of support vector X_i. X^T happens to be the input vector.

α_i > 0 for support vectors. [2].

The number of support vectors and not the dimensionality of the data determines the complexity of the learned classifier. A lower number of support vectors even when the dimensionality is too high gives a good generalization. [2].

4.5.3.2 Linearly Inseparable data

The approach used for linearly separable data holds for data that is linearly inseparable with mapping of input data to higher dimensions. Figure 3-10 shows linearly inseparable data. The input data is mapped to a higher dimensional space using a non-linear mapping and the algorithm searches for a linearly separating hyperplane in that non-linear space. The maximum margin hyperplane in the non-linear space forms a separating boundary in the original space. [2].

In solving the quadratic optimization problem of the linear SVM, the training tuples appear in the form of a dot product. Instead of computing the dot product on the transformed data, kernel function $K(X_i, X_j)$ is used on the original input making $K(X_i, X_j)$ = $\phi(X_i). \phi(X_j)$. This way all calculations are made in the input space. [2].

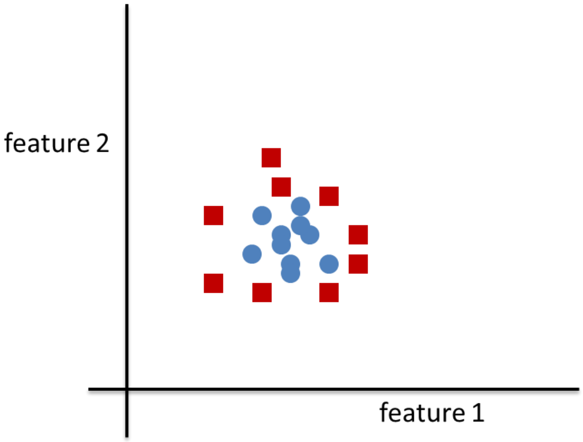


Figure 4-7 Non-linearly separable data

Some of the kernel functions we used are:

- Polynomial Kernel of degree d: $K(X_i, X_j)$ = (1 + $X_i . X_j$)
- Gaussian radial basis function kernel: $K(X_i, X_j)$ = $e^{-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}}$

4.5.3.3 SVM to Handle Outliers (The Non-separable Case)

Using a linear SVM with hard margin it would not be possible to find a solution to the problem shown in Figure 3-11, as the data is inseparable in linear space. The constraints 1 and 3 discussed earlier are relaxed by introducing slack variables ξ_i, where i = 1,…,l. The length of the training data is denoted by the variable l. [5].

The equations then become:

$$WX + b \ \geq 1 \ - \xi_i \, for \, y_i = \ +1$$

and

$$WX + b \leq -1 + \xi_i \, for \, y_i = -1$$

where ξ_i ≥ 0 $\forall_i$. In particular, ξ_i is greater than 0 for outliers and is equal to 0 when the points lie on the correct side of the margin. $\sum_i \xi_i$ is an upper bound on the number of training errors. A cost parameter C is multiplied with the sum of the errors, where, the value for the parameter C is chosen by the user. A larger C corresponds to assigning higher penalty for errors. [5].

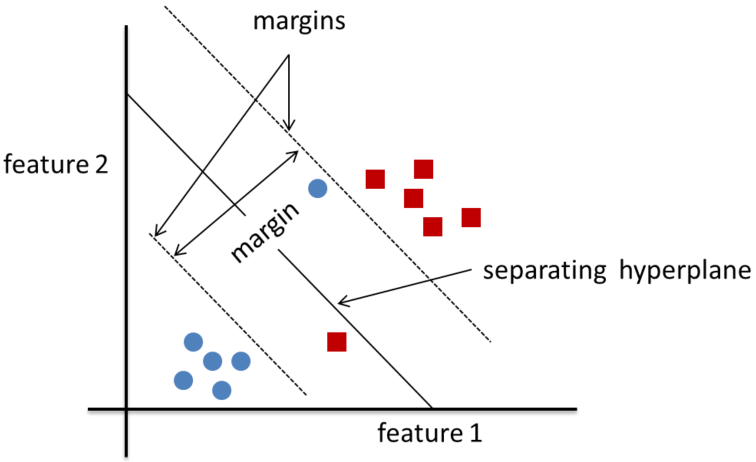


Figure 4-8 SVM using non-linearly separable data [5]

The objective function to be minimized is of the form

$$\frac{1}{2}||W||^2 + C\sum_{i}\xi_i$$

Using Lagrangian formulation and solving for the solution using Karush – Kuhn – Tucker conditions a separating hyperplane using Largange multipliers α_i is obtained:

$$\sum_{i=1}^{l} y_i\alpha_i X_i X^T + \ b_0$$

where $0 < \ \alpha_i < C$ is the additional constraint that needs to be satisfied. [5].

4.5.4 Support Vector Data Description (SVDD)

Support Vector Data Description is used for novelty detection or outlier detection. A spherically shaped decision boundary around a set of objects is constructed by a set of support vectors describing the sphere boundary. The method is used to obtain description of a set of objects. This description should cover the class of objects represented by the training set. The SVDD should ideally reject all other possible objects in the object space. [6].

A sphere with minimum volume containing most of the objects is found. To allow data points outside the sphere, slack variables $\xi_i$are introduced. Of the sphere, described by **center *a*** and **radius *R*,** the radius is minimized as

$$F(R,\ a,\ \xi_i) = R^2 + C\sum \xi_i$$

under the constraints

$$(x_i\ \ \mathrm{a})^T(x_i\ \ \mathrm{a}) \leq R^2\ +\xi_i \qquad \forall_{i,}\ \xi_i \geq 0$$

C is here a trade off between the volume of the sphere and the target objects rejected.[6].

Using Lagrangian formulation and solving for the solution using Karush – Kuhn – Tucker conditions the equation to determine if the test point lies within the hypersphere is obtained.

A test object z is accepted when

$$\left(z\ \ a\right)^T\left(z\ \ a\right) \leq\ R^2$$

Expressing the center of the sphere in terms of support vectors, a test object is accepted when

$$(\boldsymbol{z}.\boldsymbol{z}) - \mathbf{2}\sum \boldsymbol{\alpha}_{\boldsymbol{i}}(\boldsymbol{z}.\boldsymbol{x}_{\boldsymbol{i}})\ +\ \sum \boldsymbol{\alpha}_{\boldsymbol{i}}\boldsymbol{\alpha}_{\boldsymbol{j}}(\boldsymbol{x}_{\boldsymbol{i}}.\boldsymbol{x}_{\boldsymbol{j}}) \leq\ \boldsymbol{R}^{\mathbf{2}}$$

where, $0 \leq\ \alpha_i\ \leq C,\ \sum \alpha_i = 1,\ a =\ \sum_i \alpha_i\, x_i.\ \alpha_i\ and\ \alpha_j$ are the Lagrange multipliers of the support vectors [6].

Appendix A shows varying decision boundaries on banana data set using SVDD.

4.5.5 One Class Support Vector Machine (OCSVM)

One class SVM is an algorithm for novelty or outlier detection. A hyperplane in the feature space offset by a parameter ρ separates the data points from the origin. The data is separated from the origin by minimizing:

$$\frac{1}{2}\,||w||^2 + \frac{1}{vl}\sum \xi_i \text{ - } \rho$$

subject to the constraints,

$$\big(w.\phi(x_i)\big) \geq \rho - \xi_i, \qquad \xi_i \ \geq 0$$

where, $\phi(x_i)$ is the transformed feature, ξ_i is the slack variable to allow outliers. [7].

Decision Function $f(x) = \mathrm{sign}(\big(\mathrm{w}.\phi(x)\big) - \rho)$ should be positive for most of the values contained in the training set. Using Lagrangian formulation and solving for the solution using Karush – Kuhn – Tucker conditions the equation to determine if the test point lies on one side of the hyperplane representing the object space is determined by the kernel expansion which results in

$$f(x) = sign(\sum \alpha_i \, \mathrm{k}(x_i, x) - \rho)$$

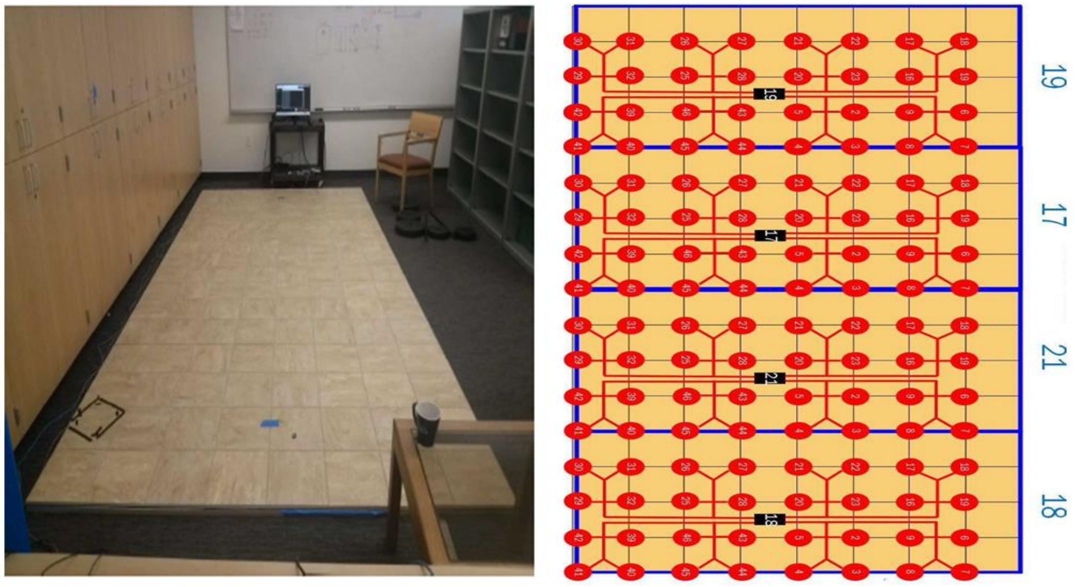
with the constraints $0 \leq \ \alpha_i \ \leq \frac{1}{vl},\ \sum \alpha_i = 1$. Here, v is the upper bound on the number of support vectors and lower bound on the number of outliers. l is the size of the training data and α_i is the support vector [7]. The value of ρ can be computed as $\sum_j \alpha_j \, \mathrm{k}(x_j, x_i)$

Chapter 5

DATA

5.1 Data Collection

For data collection we use a pressure sensitive smart floor and experimental data obtained in previous work [10]. A series of pressure monitoring sensors are here placed underneath the floor tiles to record pressure data. Pressure exerted by a subject while performing activities like standing and walking are collected at the rate of 25Hz. Data is transmitted continuously from the floor containing 128 sensors placed under 128 tiles to a nearby computer. The size of each tile is 30 cm x 30 cm. The laid out tiles form a grid of 8 x 16 tiles. Data was collected from 35 participants consisting of 11 males and 24 females. Balance and walking data were collected from each of the participants. Figure 5-1 shows the floor that generates the data and the layout of sensors underneath the floor. [10].



(i)

(ii)

Figure 5-1 (i) Is the floor that generates the data (ii) Is the layout of sensors underneath the floor [10].

5.2 Walking Trial

The subjects walked the full length of the floor twice, from the start position to the end position, at their regular pace. They then walked towards opening the cabinet that was adjacent to the floor before walking back to the stopping position. The subjects further picked up a cup placed on a table adjacent to the floor near the stopping position and continued to walk towards the start position. To complete the walking trial, the subject was asked to perform a slow walk. More information about other test and activities performed are detailed in the thesis, “Gait Analysis on a Smart Floor for Health Monitoring” [10]. Figure 5-2 shows the normal walk and slow walk trials carried out by each subject.

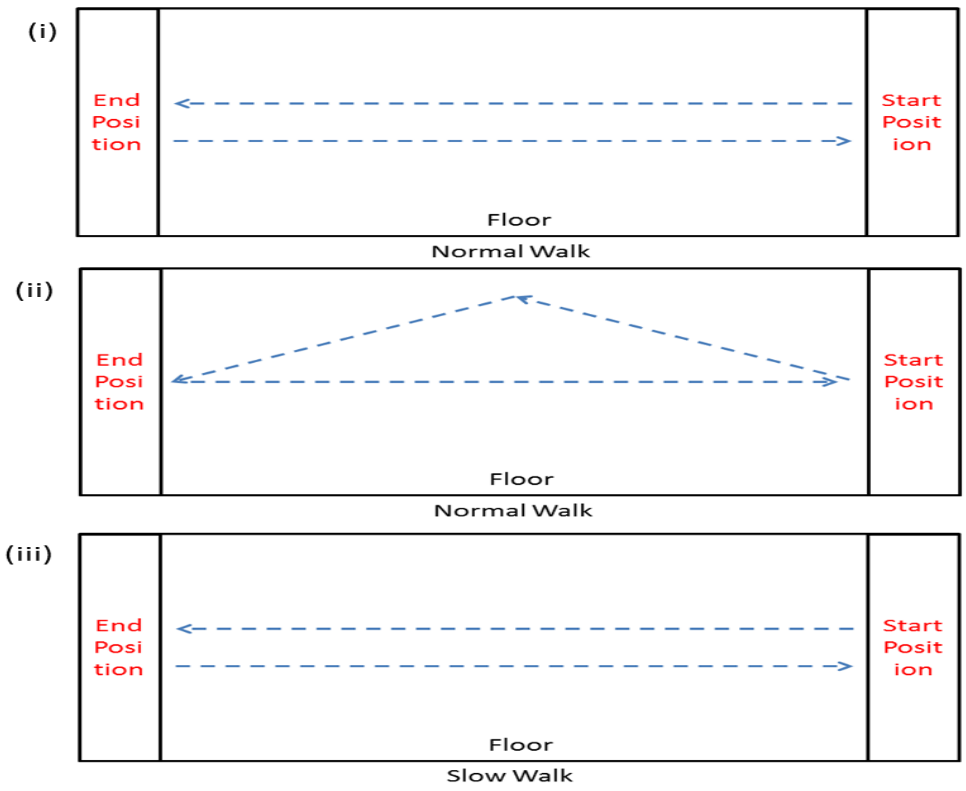


Figure 5-2 Normal and slow walk trials carried out by each subject. (i) - (ii) Is the normal walk trial. (iii) Is the slow walk trial. [10]

5.3 Data Calibration and Preprocessing

128 units of Tekscan FlexiForce A401, sensors each with an output between 0 to 1023 were used. The sensors behave linearly and adhere to the slope-intercept form w = ax + b. Where *w* is the weight in pounds that is to be determined, *x* is the raw output from the sensor and, *a* is the coefficient and b is the constant. [10].

After calibration of the data, we obtain the pressure being exerted by the subject while standing or walking on the floor along with the weight of the tile. The weight of the tile is subtracted from the data after finding mode for each sensor. This allows us to extract the pressure exerted by the subject on the floor. [10].

5.4 The Data

The data comprises the location coordinates x and y and the associated pressure value. The pressure value is determined by averaging the pressure over the region of activated sensors. Segmenting strides requires segmenting relevant walking data. Using this data as an input to a learning algorithm to classify the segments as walking or not walking would not be feasible as the location parameters are constantly varying and carry no meaning. We therefore transform the features from time domain to the frequency domain, where each feature represents amplitude at a specific frequency. The size of the time series data that needs to be transformed to frequency domain for activity recognition should be chosen in such a way that the segment should be long enough to differentiate short feet movement from walking and short enough so that multiple activities are not merged. If the length of the segment is too long there would be a high chance of the segment combining two or more activities. For example, the segment could contain data of walking activity and turning activity. Owing to the mixture of data, such segments will have to be discarded as they would result into erroneous gait parameters.

5.5 Segment Size

In the research, “gait analysis on smart floor for health monitoring” [10] one of the 30 subjects had the lowest step frequency of 1.3135 steps per second, which translates to around 78 steps per minute. Also, the number of steps per minute is on an average between 81 to 125 for individuals in the age group 65 to 80 [15]. One second of data should therefore include at least one heel strike of the walk. Therefore we determine one second of continuous data to be of sufficient size for activity recognition. Data is generated at 25Hz. One second of data contains 25 data points. 25 data points form a data segment. The data segment is a matrix of 25 X 3 values.

5.6 Periodicity

The data segment may or may not be periodic. In case of standing segments, the data could be periodic as the person is balancing the weight of the body. In case of walking the data is not periodic due to the shorter length of the segment. Data segments containing a mixture of different activities like standing and walking are non-periodic. Fourier Transform can be applied for both periodic and non-periodic data to obtain the underlying frequencies [13].

5.7 Features and Feature Transformation

The input data is a time series data segment. We compute Fourier Transform of this data. We set the segment length to 25 i.e. the number of data values that are in one segment which need to be transformed to frequency domain.

To compute Discrete Fourier Transform of the data segment we use the Fast Fourier Transform function *fftn* of Matlab to convert the data to frequency spectrum along the three dimensions of the data segment.

5.7.1 Feature Extraction

To extract walk segments we use unsupervised learning to form clusters and then use supervised learning to differentiate walk segments from the rest of the data.

Clustering

The data of each individual across 30 subjects is broken down into segments, where each segment contains 25 data points. The segments are converted from time domain to frequency domain using multidimensional Fourier Transform to obtain the frequency spectrum along each dimension. We group similar segments by clustering.

Similarity Metric for Clustering

To find similarity between two frequency spectrums we compute spectral coherence between them. This allows us to look at how coherent the two frequencies are rather than how close the magnitudes of the corresponding frequencies are. For example, if a simple signal in one dimension is multiplied by a random number, the spectral coherence between the signals is the highest, while the Cartesian distance between these two signals can be large when the multiplicative factor is a large value or a small value. Hence the approach of using Cartesian distance as a distance metric to obtain similarity between the signals is discarded. To simply put, we are interested in the signature of the frequency spectrum and using spectral coherence as a similarity measure allows us to compare the signatures of the frequencies.

Since the similarity metric is not Cartesian distance, k-means clustering approach is discarded. Agglomerative hierarchical clustering approach was used to form clusters using spectral coherence as a similarity metric on data segments of 30 subjects.

Linkage

We used the complete linkage criterion for the agglomerative hierarchical clustering because of its global closeness property [2]. Complete linkage ensures that similar segments are tightly grouped together.

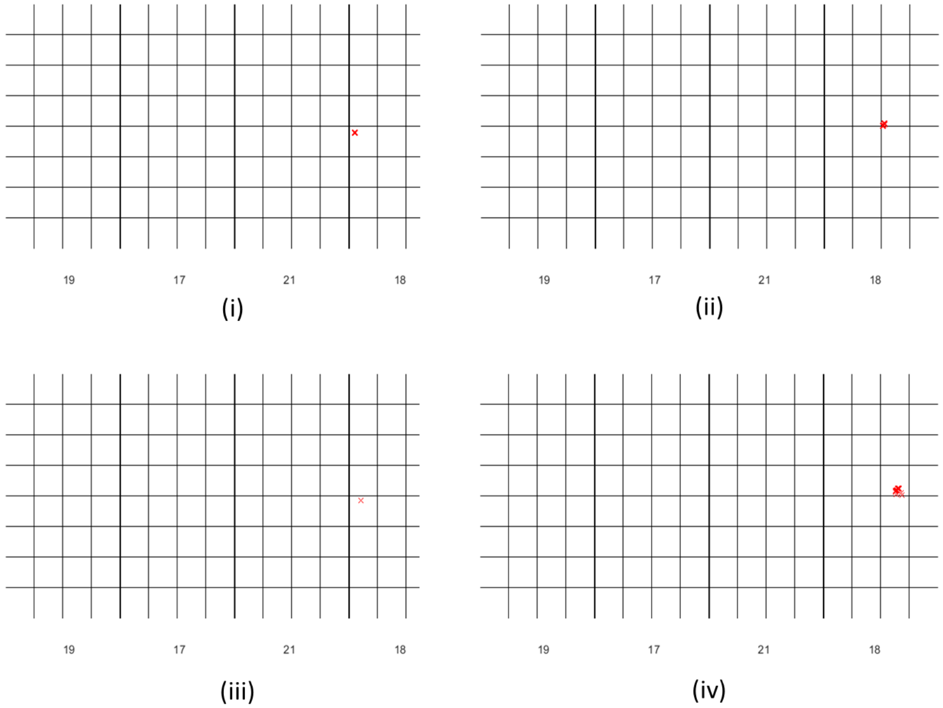
Cluster Evaluation

A simple visualization representing the floor of the lab was created to plot the location parameters of the data segments. The purity of the cluster was assessed by visualizing the plot showing the movement of the center of pressure. The labels of the clusters were assigned by visualizing the segments of the cluster.

5.8 Types of Segments and their Characteristics

5.8.1 Standing Segments

The parameters show little change in the location parameters. The plot in Figure 5-3 shows the center of pressure (COP). The center of pressure is in a highly dense region. The pressure values associated with these values are within a similar range.



(i)

(ii)

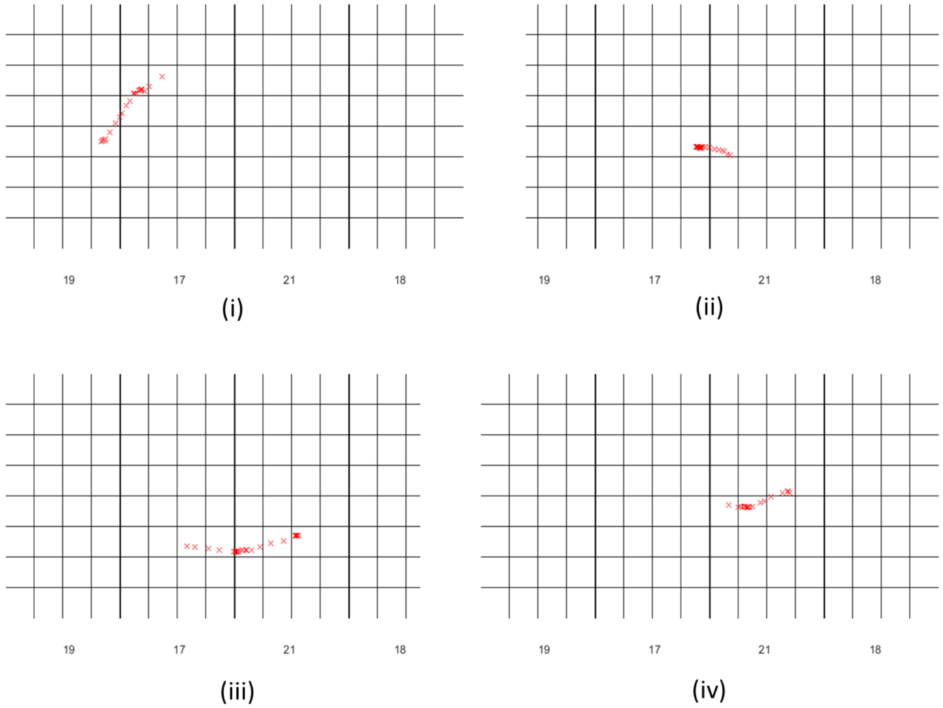
(iii)

(iv)

Figure 5-3 (i) – (iv) Standing COP trial segments.

5.8.2 Walking Segments

There are two kinds of walking segments, normal walk and slow walk segments. The normal walk resembles the regular walk of the subject while in slow walk, the length of the steps is much shorter than the regular walk segments. Also, the slow walk takes more time than normal walk. The average distance between each center of pressure (COP) point to the next COP point is more in regular walk segment compared to slow walk segment.



(i)

(ii)

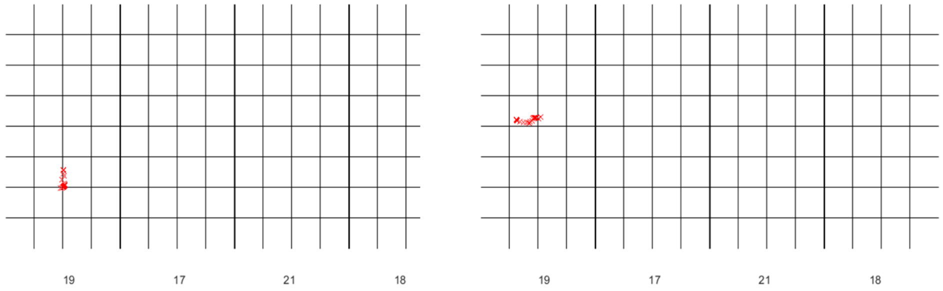
(iii)

(iv)

Figure 5-4 (i) – (iv) Walking COP trial segments.

5.8.3 Short Feet Movement or Transfer of Pressure

There are short feet movements that are similar to walk segments but much shorter in length. Their COP points look like those between standing and slow walking. Transfer of pressure from one leg to another also creates the COP plot pattern as that of short feet movement.

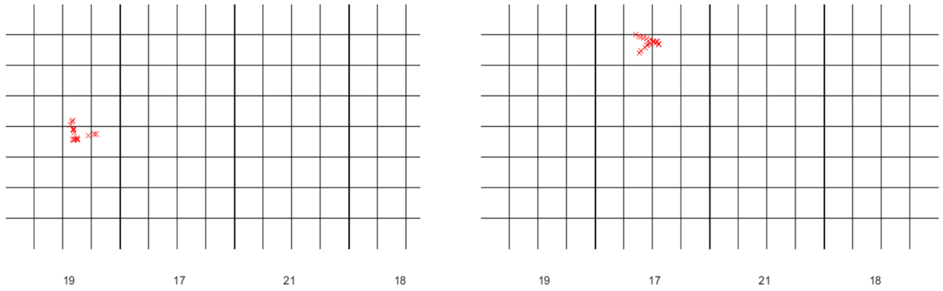


(i) (ii)

Figure 5-5 (i) – (ii) Short feet movement COP trial segments.

5.8.4 Turning Segments

Turning segments contain a loop in the COP trail. Some COP trail segments also have the shape of L during turning.



(i) (ii)

Figure 5-6 (i) – (ii) Turning COP trial segments.

5.8.5 Mixture Segments

These segments can be a combination of any of the above segments. Since the segment length has 25 rows, most often the mixture segments contain activities of two kinds as the segment is one second in length.

5.9 Cluster Formation

Clustering is performed on data of 30 subjects. The data segments include activities described above. The label of the segments is unknown and is to be assigned. A label assigned to a cluster assigns labels to each of the data segments of the cluster. The first segments to get clustered are the standing segments owing to higher coherence among them in the frequency domain. The short feet segments are similar to the standing segments and get clustered with the standing segments once the standing segments are clustered together. The walking segments are likely to get clustered based on the angle of walking. Walk segments along an angle are more likely to get clustered. Short feet segments and turning segments which are similar to walk segments are likely to get clustered. The walk segments are the ones that get clustered at the end owing to the larger variation in the frequency spectra.

5.9.1 Cluster Overlapping, Cluster Properties and Cluster Labels

Figure 5-7 shows overlap among clusters of different activities. The short feet movement and turn segments are labeled as noise class. The noise class overlaps with the standing segments and relevant walk segments. The standing segments are called the standing class. The relevant walk segments are those segments that contain walking data which is suitable for stride extraction. The relevant walk segments which include some of the noise data are labeled as the walking class while assigning labels to the clusters. Clusters contain small amount of noise.

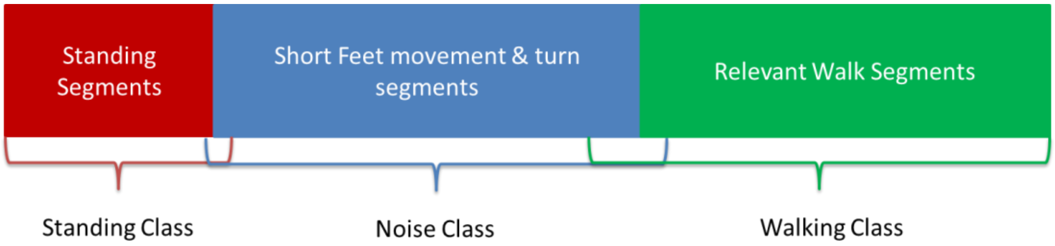


Figure 5-7 Different classes and their overlapping.

Each of the data points is an individual cluster before merging. The clusters are merged one point at a time to form the hierarchy of clusters, eventually leading to one cluster that includes all data points [2]. If this clustering is cut at the lowest level i.e. at a height where each individual point is a cluster we would have to assign labels for each point. Doing so will result in not taking advantage of the clustering process. Contrary to cutting the cluster at the lowest height, cutting the cluster at very high levels will result in a high degree of overlapping between the classes. Figure 5-8 shows cutting the cluster at the lowest level and cutting the cluster at a higher level. Cutting the cluster at the lowest level results in {A}, {B}, {C}, {D}, {E}, {F}, and {G} each being an individual cluster thus resulting in 7 clusters. Cutting the cluster at a higher level as shown in Figure 4-6 results in 3 clusters {A,B,C}, {D,E} and {F, G}.

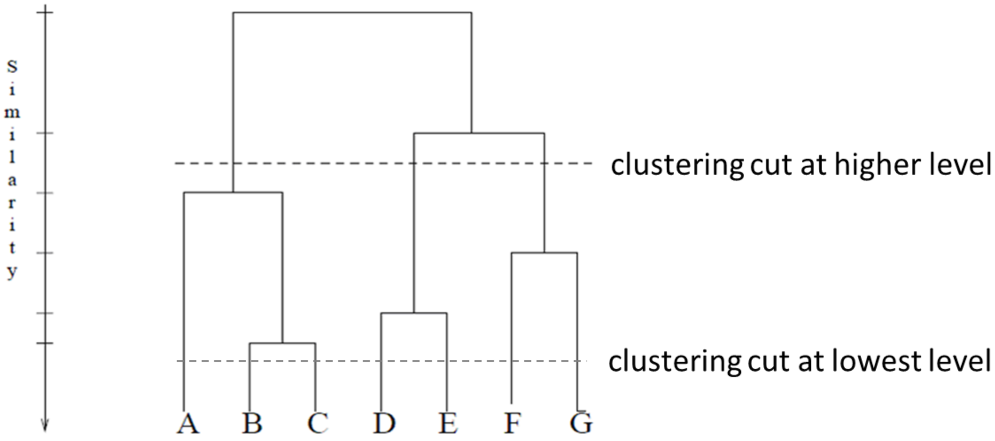

Similarity
clustering cut at higher level
clustering cut at lowest level
A
B
C
D
E
F
G

Figure 5-8 Cutting the cluster. [11]

3673 segments from 30 subjects are clustered. Cutting the cluster at a height where 80% of the points belonged to a cluster containing more than one point, resulted in the following:

- Large clusters of standing data.
- Small clusters of walking data.
- Walking cluster that exists as singleton cluster.
- Small noise clusters.
- Short feet movement overlapping with standing data.
- Short feet movement and turning segments overlapping with walking data.

The clusters were labeled as standing or walking or as noise segments by visualizing the COP trail of all the segments belonging to the cluster. Figure 5-9 shows the visualization used to assign labels to the segments. Owing to the overlap of noise class with walking class, the walking class was further separated into noise and relevant walk segments and is as shown in Figure 5-10. Relevant walk segments were further classified as start of walk, end of walk and mid walk segments. The walking class data across 30 subjects was manually labeled as noise or start of walk or end of walk or mid walk segments.

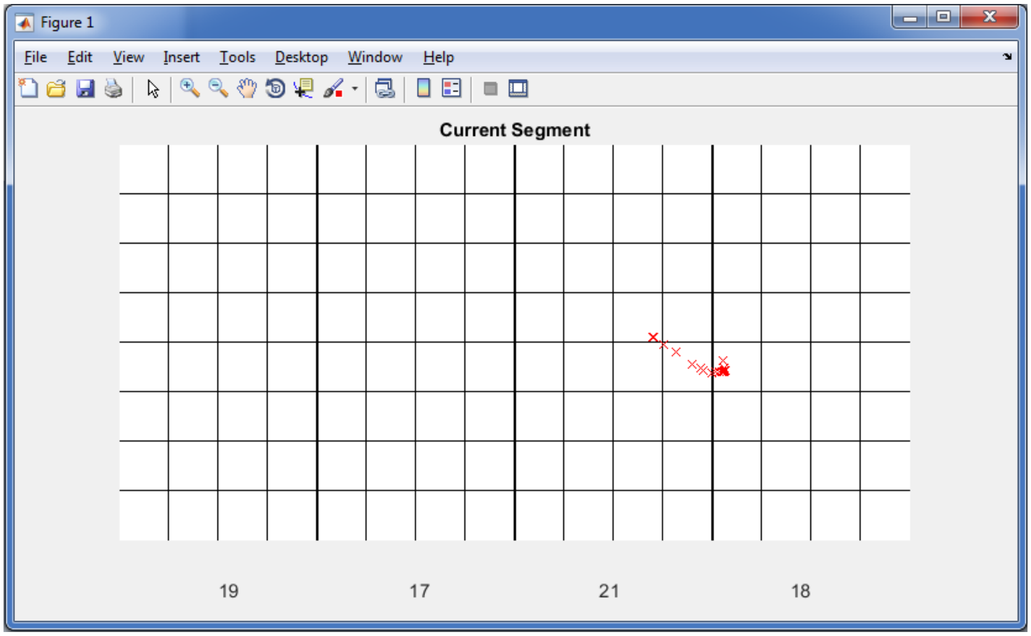


Figure 5-9 Visualization of the current segment.

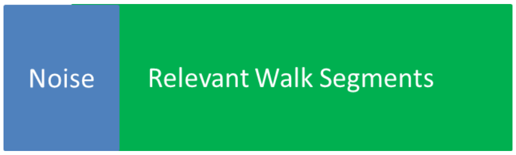


Figure 5-10 Overlap of noise and relevant walk segments

5.9.2 Labeling of Walk Segments

Using the visualization with three windows, one window showing the current segment, the other two windows showing the next segment and the previous segment, labels were assigned to the current segment. Segments that involved standing activity at first and walking activity towards the end of the segment were labeled as *startOfWalk*.

Segments that involved walking activity in the beginning of the segment and gradually having standing activity were labeled as *endOfWalk*. Based on the boundaries of the floor and prior knowledge about the activity along with visualization, *endOfWalk* labels were carefully assigned. The segments that were in between *startOfWalk* segments and *endOfWalk* segments were labeled as *midWalk* segments. Any other mixture segments that do not fit into the category described above were added to *noise* category. Figure 5-11 shows the setup to assign label for the current segment based the COP trial of the current segment, previous segment and next segment.

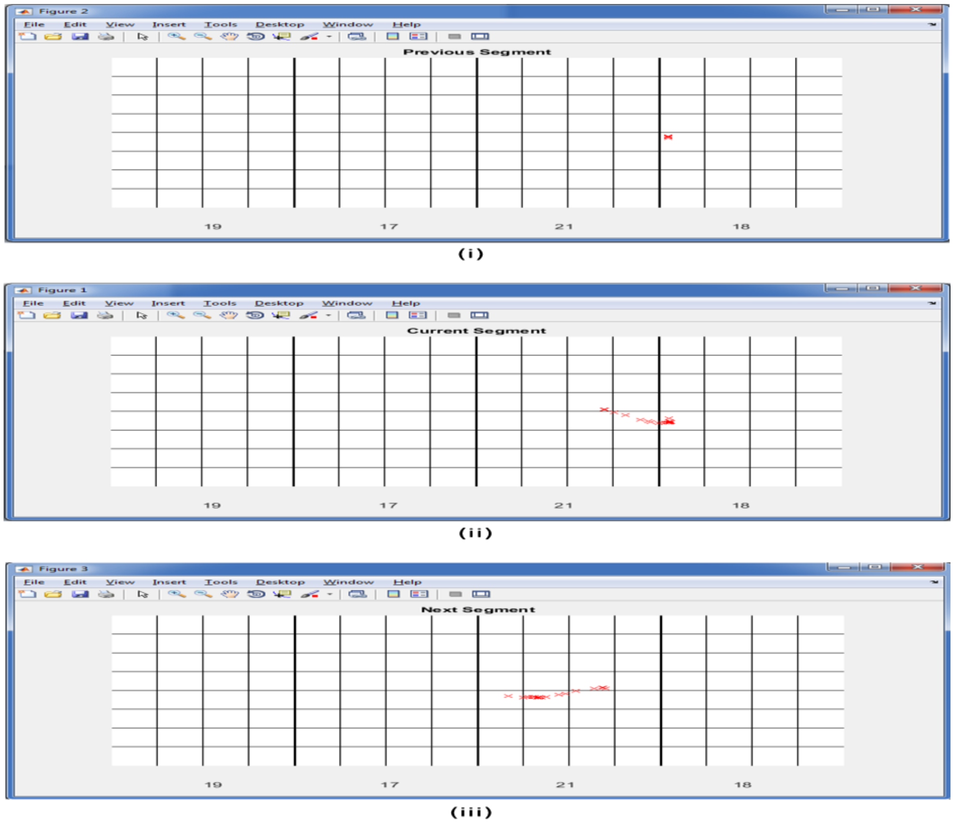


Figure 2
File Edit View Insert Tools Desktop Window Help
Previous Segment
19
17
21
18
(i)
Figure 1
File Edit View Insert Tools Desktop Window Help
Current Segment
19
17
21
18
(ii)
Figure 3
File Edit View Insert Tools Desktop Window Help
Next Segment
19
17
21
18
(iii)

Figure 5-11 Visualization of (ii) current, (i) previous and (iii) next segment used in assigning label for the current segment

5.11 Classification

The input data for classification is the frequency spectrum matrix of size 25 X 3 converted into a vector. Classification is carried out in two stages. At stage one we classify the frequency spectrum as belonging to either one of the classes: standing, noise or walking. When stage one classifies the result as walking, we use stage two classification to determine if the walk segment is relevant. At stage two of classification, we separate walk segments from noise. We also used stage two data to classify relevant walk segments as startOfWalk segments, *endOfWalk* segments, *midWalk* segments from noise segments. Since beginning of walk and end of walk segments usually are not correct representation of one's walking, they could be discarded when sufficient mid walk segments are available. MidWalk segments are a more accurate representation of the person's walk.

The classifiers used at stage one are Decision Tree and Gaussian Naïve Bayes Classifiers. The classifiers used in stage two are Decision Tree, Linear Support Vector Machines and Gaussian Naïve Bayes. To classify relevant walk segments as startOfWalk segments, *endOfWalk* segments, *midWalk* segments from noise segments, we used Decision Tree and Gaussian Naïve Bayes Classifier.

The training data for classification was the data belonging to 15 subjects and the testing was carried out on the data belonging to the remaining 15 subjects which was not used during training. Though it is possible to determine if the segment is an end of walk segment or beginning of walk segment based on the predicted class of the next segment or the previous segment, creating classes of data as startOfWalk, endOfWalk and midWalk to perform classification was to see if the segments could be accurately classified in the frequency domain. The results of using the classifiers such as decision tree, Gaussian Naïve Bayes and Linear SVM on start and end segments were between

60% – 70% and hence it was not used. Also, leaving out startOfWalk and endOfWalk segments will reduce partially relevant walk segments for feature extraction.

5.12 Stride Extraction

In this research we extract gait parameters using the methods from [10] based on the initial contact (heel strike) time and toe off time. The heel strike and toe off times are found using the center of pressure speed. The center of pressure speed has a peek value when a heel strike occurs. The toe off occurs when the center of pressure speed drops. With this information strides can be determined. The double support – single support detection algorithm [10] is used in extracting strides. Figure 5-12 shows a floor with COP trail when a person is walking. It shows the heel strike and toe off time. The location of heel strike on the floor is where single support takes place.

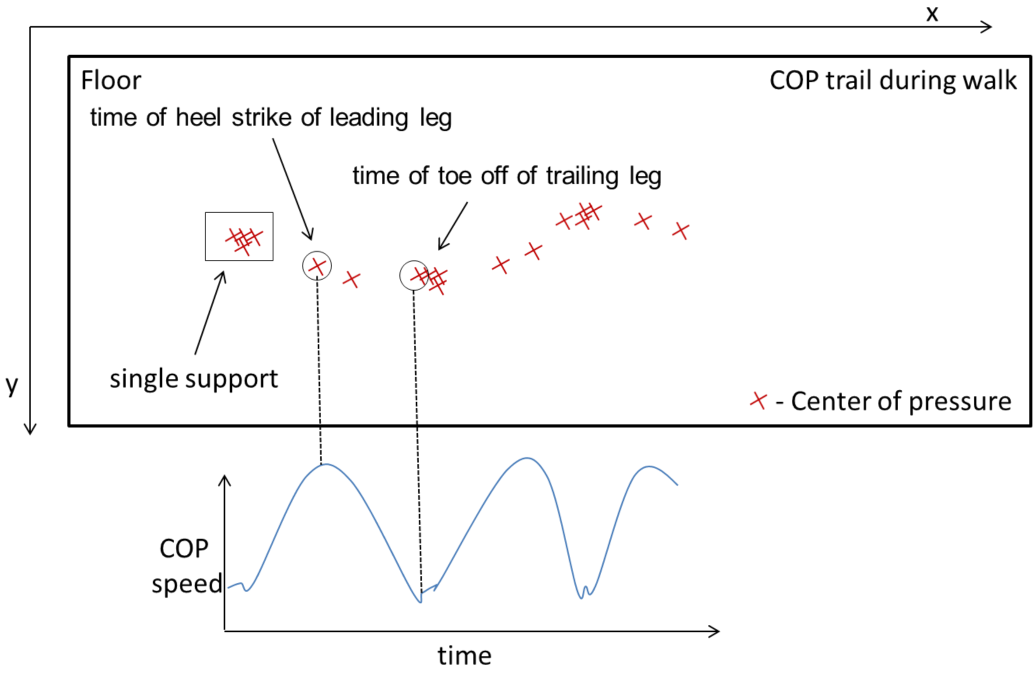


Figure 5-12 COP trail, heel strike time, toe off time and COP Speed.

After extracting relevant walk segments, such segments are passed as an input to the double support – single support detection algorithm [10] to identify strides. On obtaining the heel strike and the toe off times the strides are determined for gait parameter extraction. The strides overlap with previous strides if there are more than three heel strikes. We extract the gait parameters from these overlapping strides as shown in Figure 5-13.

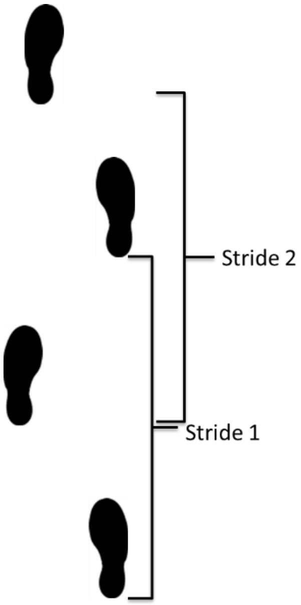


Figure 5-13 Overlapping strides for gait parameter extraction

5.13 Features

The features are extracted from the strides using the algorithms detailed in the research “Gait analysis on a smart floor for health monitoring” [10]. Figure 5-14 shows some of the spatial features that are extracted from the stride. The following features are extracted from the stride:

1. Stride length
2. Step length - 1

3. Step length – 2

4. Distance between heel strike of the leading foot and toe off of the trailing foot - 1

5. Distance between heel strike of the leading foot and toe off of the trailing foot – 2

6. Average step length

7. Step time - 1

8. Step time - 2

9. Single support time - 1

10. Single support time -2

11. Step frequency

12. Average speed

13. Feet

14. & 15. Step length ratios

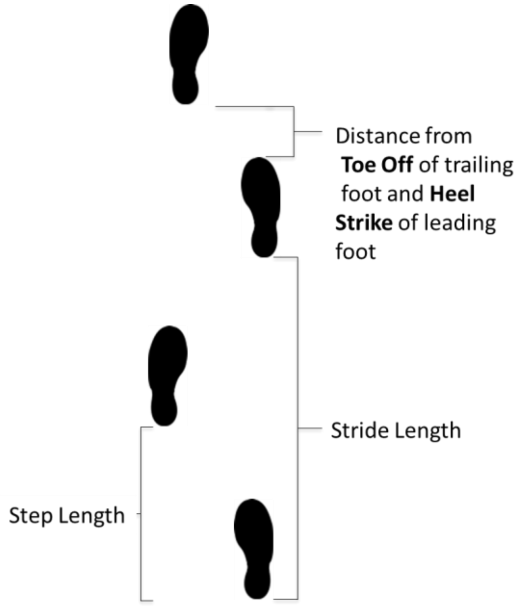

Distance from **Toe Off** of trailing foot and **Heel Strike** of leading foot
Stride Length
Step Length

Figure 5-14 Features of a stride.

Foot

Determining the leading foot is necessary to separate the step length made by the right foot from that of the left foot during the stride. The foot can be determined using the stride vector and the step vector. Considering the stride vector to be in the direction of walk, the foot can be identified by taking the normal of the stride vector and computing the dot product of the stride vector and the step vector. A positive result denotes one foot while the negative result denotes the other foot. [10].

Feature Ordering

The features extracted contain the foot parameter which indicates the leading foot of the stride. If the foot parameter is one it indicates one of the two feet and when it is negative it is the other feet. The distance and the time parameters need to be ordered such that each feature column represents the data of the particular foot. The features need to be swapped so the data is consistent with each foot. The features swapped are step lengths, step times, step ratios, single support times, distances from heel strike of the leading foot and trailing foot.

5.14 Model Building

We build personalized models for each individual for person identification and anomaly detection.

The models built for person identification are:

- One vs Rest Models
  1. Linear Support Vector Machine
  2. Support Vector Machine with RBF kernel
  3. Support Vector Machine with Polynomial Kernel – Order 2

4. Support Vector Machine with Polynomial Kernel – Order 3

- One Class Models

5. Support Vector Data Descriptor (SVDD) using RBF kernel

6. One Class Support Vector Machine(OCSVM) using RBF kernel

The models built for anomaly detection are:

- One Class Models

1. Support Vector Data Descriptor (SVDD) using RBF kernel
2. One Class Support Vector Machine (OCSVM) using RBF kernel

In one versus rest models the training data used to build a model for an individual comprises data belonging to both, the individual and the data belonging to other individuals. The data belonging to the individual is the target class data and the data belonging to all other individuals is the outlier class data. The built model is tested against the target class data and the outlier class data.

For a one class model, the training data comprises data belonging only to the individual i.e. the target class data [6][7]. The data belonging to other individuals is not used in building the model. The built model is tested against the target class data and outlier class data.

5.15 Synthetic Data Augmentation

For most subjects, the number of strides extracted that belong to normal walk is between 15 to 25. A feature vector is extracted from each stride. One of the subjects among the 30 subjects had only 5 strides. As a result, synthetic data was generated for the subject based on the existing strides using interpolation techniques. 10 newly generated points were added to the individual’s existing set to increase the stride count of normal walking data to 15.

In building models for individuals, two thirds of the data was used for training and one thirds of the data was used for testing. The number of features in the feature vector is 14. The number of feature vectors for each of the subjects that was used for training was between 10 to 16. Building the model on less training data and more features resulted in overfitting. Overfitting could be established by lower training errors and higher test errors [4].

For the reasons mentioned above, synthetic data generation was necessary. The algorithm for synthetic data generation using linear interpolation of existing data points is as detailed below.

Algorithm 1: Synthetic Data Generation

Input:

- **originalData**: Gait parameters extracted from strides belonging to a subject.
- **percentOfPoints**: Percentage of points having at least one neighbor.
- **noOfPointsToGenerate**: No of synthetic data points to generate.

Output:

- **syntheticData**: Generated synthetic data.

```
    distanceMatrix = ComputeCartesianDistanceBetweenPoints(data);
    lowerTriangleDistanceMatrix =
SetDiagonalAndUpperTriangleToMax(distanceMatrix);

    % sortedDistance is the sorted distance of the lower triangle

    % xVector contains corresponding x value of the sorted distance

    % yVector contains corresponding y value of the sorted distance
```

```
[sortedDistance, xVector, yVector] = …
       SortDataByLowestDistance(lowerTriangleDistanceMatrix);
noOfPoints = size(originalData,1);
noOfPointsWithAtLeastOneNeighbor = ceil (noOfPoints * percentOfPoints);
dataGenerationPoints = [ ];
while (length (dataGenerationPoints) < noOfPointsWithAtLeastOneNeighbor)
          for i=1 to length (sortedDistance)
                    % add the points according to the lowest distance
                    dataGenerationPoints = [dataGenerationPoints; xVector(i);
yVector(i)];
                    % remove duplicate points
                    dataGenerationPoints = Unique(dataGenerationPoints);
          end
end

cutOffDistance = sortedDistance(x(i),y(i));
originalAndSynData = originalData;

while (originalAndSynData  < noOfPointsToGenerate +…
          Number_Of_Rows(originalData))

          % randomly choose two different points from the data set

          % originalAndSynData and creates a new data point which is the sum of

          % alpha times the vector of one point  and 1 - alpha times the vector of

          % another point. The new data point can be created only if the distance

          % between the two points are less than or equal to the cutOffDistance.



          dataPoint1 = randi(size(originalAndSynData,1));
          dataPoint2 = randi(size(originalAndSynData,1));

          if dataPoint1 == dataPoint2
                    continue;
          end
          alpha = rand;
          beta = 1 – rand;

          newPoint = alpha*originalAndSynData(dataPoint1,:)  +
              beta*originalAndSynData(dataPoint2,:);
          %Augment new Point
```

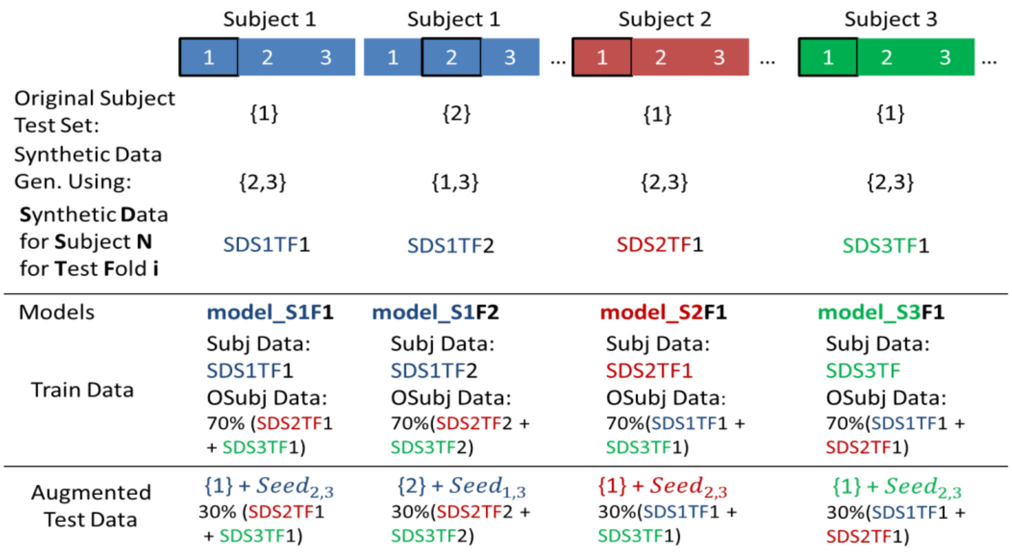
```
                originalAndSynData = [originalAndSynData; newPoint];
        end
        syntheticData = RemoveOriginalData(originalAndSynData, originalData);

        return syntheticData;
end
```

K – fold averaging using One Versus Rest SVM Models for Person Identification

The feature vectors of normal walk of an individual were randomly sampled without replacement to create three folds of data. The number of feature vectors per individual was in the range 15 – 25. When the number of feature vectors is 15, the size of the training data would be 10 and the size of the validation data would be 5. Increasing the number of folds would drastically reduce the number of data points in the validation set. Decreasing the number of folds to 2 would decrease the training data size. Hence, 3 fold validation was the better choice.

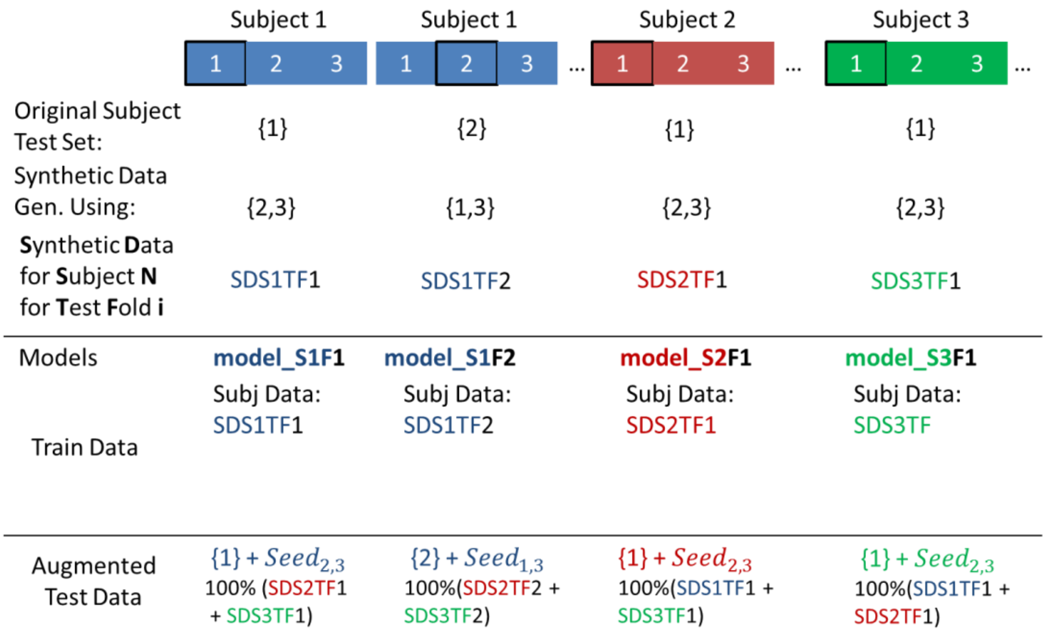
After dividing the data points into three folds, two folds were used in synthetic data generation and the other fold was used for testing. 100 data points were synthetically generated using the two folds that was reserved for synthetic data generation. Testing was carried out without augmenting the test set and with augmenting the test set. Augmented test set contains data points that were used in synthetic data generation in addition to the data points of the fold reserved for testing. The data points used in synthetic data set generation are not used for training. All other points that were not used in synthetic data generation were used for training the model. If folds 2 and 3 are used in data generation fold 1 is used for testing. If for subject 1 the synthetic data is generated using fold 2 and 3, fold 1 is used for testing. Synthetic data generated in this case is called **S**ynthetic **D**ata for **S**ubject **1** for **T**est **F**old **1**. This is abbreviated as SDS1TF1 and is shown in Figure 5-15. Similarly, SDS1TF2 is synthetic data for subject 1 for test fold 2. Figure 5-15 shows the synthetic data for subject 2 and subject 3 for test fold 1. A model is built for each individual for each fold of test set. The training data also includes the other subject data also referred to as the outlier data. The outlier data from different subjects is from the folds that generated the data for the individual. For example, when a model is built for a subject using data generated from folds 2 and 3, the outlier data of the remaining subjects must also come from the same folds. This is done so that the models are not trained using the test set of other individuals as outlier class data. 70% of the outlier data is used for training and 30% is used for testing.

Figure 5-16 shows the training data and test data for one class classifier. The only difference from the one-versus-rest approach is that the training data does not include data of other individuals. The test data however includes the data of other individuals. Since RBF kernel is used for the SVDD and OCSVM, grid search technique [12] is used to find the optimal values for gamma and Nu. Gamma is the width parameter of the Gaussian and Nu is the fraction of outliers in the original data. Nu is varied between 0.01 – 0.2 in increments of 0.01. Gamma is varied between 5 – 8. Increasing gamma increases the decision boundary. The increase in the decision boundary with increase in the width parameter gamma was seen earlier in the technical section. For lower values of the gamma parameter, the decision boundary does not fit the target data correctly. Many points that belong to the target data are omitted by the decision boundary owing to the decision boundary formed around the denser regions of the target data. The optimal values for the parameters gamma and Nu are chosen based on the lower average of target train error and target test error. Target train error is the error on the subject training data and target test error is the error on the subject test data.



	Subject 1	Subject 1	Subject 2	Subject 3
	1 2 3	1 2 3 …	1 2 3 …	1 2 3 …
Original Subject Test Set:	{1}	{2}	{1}	{1}
Synthetic Data Gen. Using:	{2,3}	{1,3}	{2,3}	{2,3}
Synthetic Data for Subject N for Test Fold i	SDS1TF1	SDS1TF2	SDS2TF1	SDS3TF1
Models	model_S1F1	model_S1F2	model_S2F1	model_S3F1
Train Data	Subj Data: SDS1TF1 OSubj Data: 70% (SDS2TF1 + SDS3TF1)	Subj Data: SDS1TF2 OSubj Data: 70%(SDS2TF2 + SDS3TF2)	Subj Data: SDS2TF1 OSubj Data: 70%(SDS1TF1 + SDS3TF1)	Subj Data: SDS3TF OSubj Data: 70%(SDS1TF1 + SDS2TF1)
Augmented Test Data	$\{1\} + Seed_{2,3}$ 30% (SDS2TF1 + SDS3TF1)	$\{2\} + Seed_{1,3}$ 30%(SDS2TF2 + SDS3TF2)	$\{1\} + Seed_{2,3}$ 30%(SDS1TF1 + SDS3TF1)	$\{1\} + Seed_{2,3}$ 30%(SDS1TF1 + SDS2TF1)

Figure 5-15 Training data and test data for one-versus-rest classifier.



	Subject 1	Subject 1	Subject 2	Subject 3
	1 2 3	1 2 3 …	1 2 3 …	1 2 3 …
Original Subject Test Set:	{1}	{2}	{1}	{1}
Synthetic Data Gen. Using:	{2,3}	{1,3}	{2,3}	{2,3}
Synthetic Data for Subject N for Test Fold i	SDS1TF1	SDS1TF2	SDS2TF1	SDS3TF1
Models	model_S1F1	model_S1F2	model_S2F1	model_S3F1
Train Data	Subj Data: SDS1TF1	Subj Data: SDS1TF2	Subj Data: SDS2TF1	Subj Data: SDS3TF
Augmented Test Data	$\{1\} + Seed_{2,3}$ 100% (SDS2TF1 + SDS3TF1)	$\{2\} + Seed_{1,3}$ 100%(SDS2TF2 + SDS3TF2)	$\{1\} + Seed_{2,3}$ 100%(SDS1TF1 + SDS3TF1)	$\{1\} + Seed_{2,3}$ 100%(SDS1TF1 + SDS2TF1)

Figure 5-16 Training data and test data for one class classifier.

Classification

We use the test data belonging to a fold *i* of the subject and use models across all subjects that have been trained with data other than fold *i* to predict the number of test points that belong to the model. Figure 5-17 explains the process. The results of the k – folds across all models are averaged to obtain the average across the folds of the probability that the data belongs to the particular individual. The test data includes augmented and un-augmented data sets. Augmented data set is as described earlier. In Figure 5-15 the data belongs to subject 1. The figure consists of 3 subjects and 3 models for each subject. For each of the folds belonging to the subject, after classifying using all models belonging to the fold, we average the results to get the probability of the data belonging to an individual. The data is said to belong to a subject based on the probability values across all models of all subjects on the subject's data. It is predicted that the data belongs to a particular individual based on the highest probability. Figure 5-17 shows that the data belongs to subject 1 with a probability of 0.62. Here, the data used to test belongs to subject 1. This approach is used for both, one-versus-rest classifiers and one class classifiers.

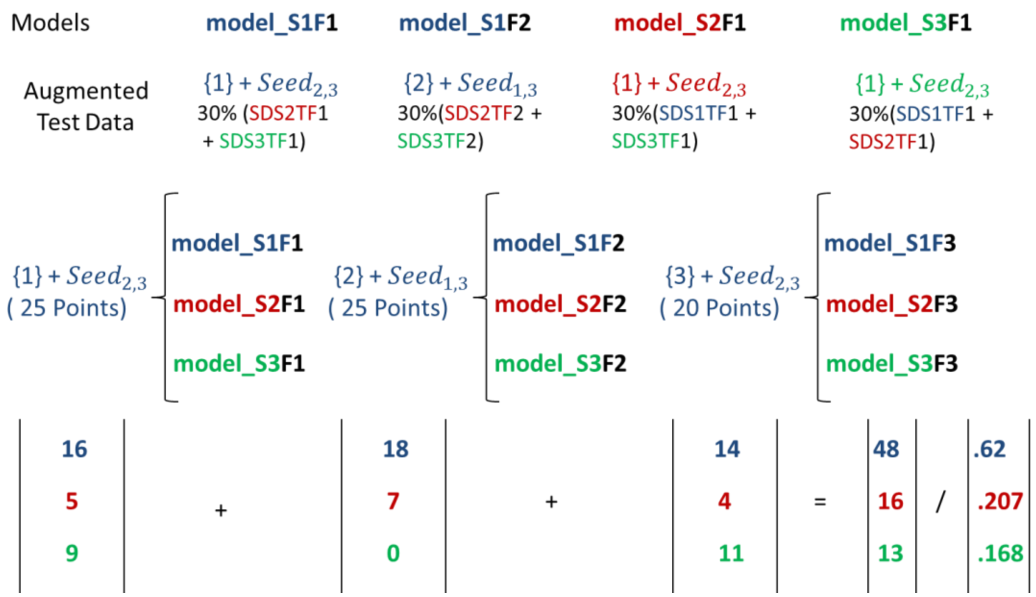


Figure 5-17 Averaging predictions across all models per fold for person identification for one-versus-rest classifier.

Chapter 6

RESULTS AND ANALYSIS

6.1 Classification at Stage 1

At this stage we classify input data as Noise or Standing or Walking segments. The classifiers used to separate Noise, Standing and Walking segments are Gaussian Naïve Bayes and Decision Tree. Of the 30 subjects, the data of 15 subjects were used in building the model. The classifiers were tested on the untrained 15 other subjects. The classification is performed on both, training data and testing data. Table 6-1 and 6-2 gives classification results using Gaussian Naïve Bayes Classifier on Training and Testing Data.

Table 6-1 Classification Results on Training Data

Gaussian Naïve Bayes Classifier – Training Data				
		Actual Class		
		Noise	Standing	Walking
Predicted Class	Noise	167	94	138
	Standing	40	1064	0
	Walking	8	0	441

Table 6-2 Classification Results on Testing Data

Gaussian Naïve Bayes Classifier – Testing Data				
		Actual Class		
Predicted Class		Noise	Standing	Walking
	Noise	149	76	98
	Standing	21	942	0
	Walking	8	0	428

The classification between standing and walking on training data and testing data using Gaussian Naïve Bayes classifier as shown above results in zero false positives and zero false negatives i.e., none of the walking data is classified as standing data and none of the standing data is classified as walking data. Walking data not being classified as standing data ensures that, data is not lost as a result of misclassification. Standing data not being classified as walking data ensures spurious gait parameters are not extracted. A high percentage of walking data, about 23.8% and 18.6% belonging to training data and test data respectively is classified as noise data. Standing data being classified as noise or noise data being classified as standing data does not affect our requirement. A low percentage of data, about 4.57% and 5% belonging to training data and test data respectively is classified as walking data. These segments are more similar to walking data. Such segments have to be in sequence for spurious gait parameters to be extracted. Such a sequence should have at least three heel strikes to be considered for stride extraction. For a COP speed point to be called as a heel strike the value should be greater than the average of all the COP speed values for that segment. This makes the chances of extracting parameters from noise data very low. Standing data being classified as noise or noise data being classified as standing data does not affect our requirement.

Table 6-3 and 6-4 shows the result of classification using Decision Tree.

Table 6-3 Classification Results on Training Data

Decision Tree Classifier – Training Data				
		Actual Class		
		Noise	Standing	Walking
Predicted Class	Noise	210	4	4
	Standing	1	1154	0
	Walking	4	0	575

Table 6-4 Classification Results on Testing Data

Decision Tree Classifier – Testing Data				
		Actual Class		
Predicted Class		Noise	Standing	Walking
	Noise	129	25	31
	Standing	16	993	0
	Walking	32	0	495

The classification between standing and walking on training data and testing data using Decision Tree classifier results again in zero false positives and zero false negatives. A low percentage of walking data, about 0.69% and 5.9% belonging to training data and test data respectively is classified as noise data. About 1.8% and 19.87% of noise data belonging to training data and test data respectively is classified as walking data.

Even though the percentage of noise data being classified as walking data is high with Decision Tree as compared to Gaussian Naïve Bayes classifier, we use Decision Tree as our model as 5.9% of walking data is classified as noise while the Gaussian Naïve Bayes classifier classifies 18.6% of the data to be noise when using test data. Extracting gait parameters from noise data is less likely based on the above discussion. Also, since we use another classifier at stage two to differentiate between noise and relevant walking segments, choosing Decision Tree over Gaussian Naïve Bayes classifier ensures we have more walking segments for gait parameter extraction.

Table 6-5 shows the overall accuracy using Naïve Bayes and Decision Tree Classifiers on training data and test data.

Table 6-5 Overall Accuracy of Classifiers

Classifiers	Accuracy	
	Train	Test
Naïve Bayes	85.655738%	88.262638%
Decision Tree	99.334016%	93.957002%

6.2 Classification at Stage 2

At stage 2 we classify relevant walking segments for gait parameter extraction from noise. The classifiers used in this stage are Decision Tree, Gaussian Naïve Bayes and Linear Support Vector Machines. Of the 30 subjects, data involving walking and noise segments of the 15 subjects were used in building the model. The classifiers were tested on the data of untrained 15 subjects. Classification was performed on both, training data and testing data. Table 6-6 and 6-7 gives classification results using Gaussian Naïve Bayes Classifier on Training and Testing Data.

Table 6-6 Classification Results on Training Data

Gaussian Naïve Bayes Classifier – Training Data			
		Actual Class	
Predicted Class		Noise	Walking
	Noise	70	149
	Walking	11	349

Table 6-7 Classification Results on Testing Data

Gaussian Naïve Bayes Classifier – Testing Data			
		Actual Class	
Predicted Class		Noise	Walking
	Noise	90	65
	Walking	12	359

Using Gaussian Naïve Bayes Classifier, the number of noise points misclassified as relevant walk segments is 13.5% and 11.7% for training and test data, respectively. The percentage of relevant walk segments being classified as noise is 29.9% and 15.3% for training and test data, respectively.

Table 6-8 and 6-9 gives classification results using Decision Tree for training and testing data.

Table 6-8 Classification Results on Training Data

Decision Tree Classifier – Training Data			
		Actual Class	
Predicted Class		Noise	Walking
	Noise	74	3
	Walking	7	495

Table 6-9 Classification Results on Testing Data

Decision Tree Classifier – Testing Data			
		Actual Class	
Predicted Class		Noise	Walking
	Noise	53	17
	Walking	49	407

Using Decision Tree Classifier, the number of noise points misclassified as relevant walk segments is 8.6% and 48% for training and test data respectively. The test error on the noise points being classified as walking is very high which makes it not suitable for classification. The percentage of relevant walk segments being classified as noise is 0.6% and 4% for training and test data respectively which is very low.

Table 6-10 and 6-11 gives classification results using Linear Support Vector Machines for training and testing data.

Table 6-10 Classification Results on Training Data

Linear Support Vector Machines – Training Data			
		Actual Class	
Predicted Class		Noise	Walking
	Noise	66	16
	Walking	15	482

Table 6-11 Classification Results on Testing Data

Linear Support Vector Machines – Testing Data			
		Actual Class	
Predicted Class		Noise	Walking
	Noise	73	8
	Walking	29	416

Using Linear Support Vector Machines, the number of noise points misclassified as relevant walk segments is 18.6% and 28% for training and test data, respectively. The percentage of relevant walk segments being classified as noise is 3.2% and 1.8% for training and test data, respectively. Decision Tree has 48% of noise points being misclassified as walk during testing while Gaussian Naïve Bayes classifier has 11.7% and Support Vector Machine (SVM) has 28% misclassifications. Owing to the high percentage of errors in classifying noise segments, decision tree is not considered for classification. Gaussian Naïve Bayes (GNB) classification has 15.3% of walk points being misclassified as noise during testing while SVM has 1.8% of misclassifications. Though the percentage of noise points being misclassified as walk is relatively higher using SVM compared to GNB, as mentioned earlier, the chances of extracting gait parameters from noise data is less. Hence we use SVM over GNB for classification.

Table 6-12 shows the overall accuracy of train and test data using different classifiers at stage 2 of classification.

Table 6-12 Overall accuracy of train and test data using different classifiers.

Classifiers	Stage 2 (Separate Noise and Walk Segments)	
	Train	Test
Naïve Bayes	72.366149%	85.361217%
Decision Tree	98.272884%	87.452471%
Linear SVM	94.645941%	92.965779%

6.3 Impurity in clusters

Clusters have impurity in them based on the height at which the hierarchical clustering is cut. As shown in Figure 5-8, if the clustering is cut at a lower height, the clusters obtained will be more pure compared to clusters cut at a higher height. Purity refers to the similarity of the points in the cluster. The more similar the points within the cluster are, the higher the purity of the cluster. The impurity of the cluster is high when cut at a higher level. Impurity is the lowest when all clusters are singleton clusters. Impurity is the highest when all points belong to a single cluster.

The hierarchical clustering is cut at 80% of the height. Some of the clusters have impurity in them. Noise points are part of standing and walking segments. Walk segments also have noise segments. The impurities are low and they introduce noise when the whole cluster is assigned a label. The results of clustering are therefore not exact results but are a good approximation. Therefore, when classifying, the model might be correctly classifying the data when the point is an outlier belonging to a wrong cluster. This could mean that the classification accuracies are better than what is observed.

6.4 Person Identification

Two kinds of classifiers are used for person identification, one-versus-rest classifiers and one class classifiers. The one-versus-rest classifiers are Linear SVM, SVM with polynomial kernel of order 2, SVM with polynomial kernel of order 3 and SVM with RBF kernel. The one class classifiers are Support Vector Data Descriptor with RBF Kernel and One Class SVM with RBF Kernel. Tables 6-14 to 6-16 show accuracy of training and testing for 3 subjects across 3 folds and 4 classifiers. The target test data includes seeds that generated the synthetic data. The models are not trained with these seeds. The models built here are the ones described in Figure 5-15.

Table 6-14 Accuracy for subject 1 across 3 folds using one-versus-rest classifier

		Target Train Data	Target Test Data	Outlier Train Data	Outlier Test Data
Fold 1	poly -2	100%	56.521%	100%	91.851%
	poly -3	100%	73.913%	100%	88.518%
	RBF	93.333%	34.782%	100%	100%
	Linear	86.666%	26.086%	99.206%	85.555%
Fold 2	poly -2	100%	65.217%	100%	96.296%
	poly -3	100%	60.869%	100%	100%
	RBF	99.047%	56.521%	100%	100%
	Linear	80%	39.130%	99.047%	99.259%
Fold 3	poly -2	100%	56.521%	100%	97.407%
	poly -3	100%	56.521%	100%	100%
	RBF	98.095%	34.782%	100%	100%
	Linear	95.238%	43.478%	97.460%	100%

Table 6-15 Accuracy for subject 2 across 3 folds using one-versus-rest classifier

		Target Train Data	Target Test Data	Outlier Train Data	Outlier Test Data
Fold 1	poly -2	100%	61.904%	100%	95.555%
	poly -3	100%	66.666%	100%	100%
	RBF	99.047%	57.142%	99.682%	100%
	Linear	96.190%	80.952%	99.365%	100%
Fold 2	poly -2	100%	66.666%	100%	97.037%
	poly -3	100%	66.666%	100%	96.296%
	RBF	100%	66.666%	100%	99.629%
	Linear	80%	52.380%	99.206%	96.666%
Fold 3	poly -2	100%	81.818%	100%	99.259%
	poly -3	100%	63.636%	100%	100%
	RBF	99.038%	59.090%	100%	100%
	Linear	94.230%	63.636%	99.682%	99.629%

Table 6-16 Accuracy for subject 3 across 3 folds using one-versus-rest classifier

		Target Train Data	Target Test Data	Outlier Train Data	Outlier Test Data
Fold 1	poly -2	100%	78.571%	100%	98.888%
	poly -3	100%	85.714%	100%	98.518%
	RBF	98.058%	64.285%	100%	98.888%
	Linear	98.058%	92.857%	100%	97.777%
Fold 2	poly -2	100%	57.142%	100%	95.185%
	poly -3	100%	64.285%	100%	97.777%
	RBF	99.029%	50%	100%	98.888%
	Linear	99.029%	64.285%	100%	88.518%
Fold 3	poly -2	100%	71.428%	100%	98.518%
	poly -3	100%	78.571%	100%	97.777%
	RBF	100%	78.571%	100%	98.888%
	Linear	100%	85.714%	100%	97.037%

Using the models and the approach described in Figure 5-17 we predict the subjects to whom the data belongs after validation. Table 6-17 and 6-18 show the correct number of predictions without the seeds included in the test data and with the test data including the seeds respectively.

Table 6-17 Correctly predicted subjects on test data not containing seeds. Predictions made using validation for each of the 4 SVM classifiers.

No of Subjects	Poly - 2	Poly – 3	RBF	Linear
	Correctly Predicted Subjects			
First 10	8	6	7	5
First 20	12	10	12	7
First 30	16	15	11	8

Table 6-18 Correctly predicted subjects on test data containing seeds. Predictions made using validation for each of the 4 SVM classifiers.

No of Subjects	Poly - 2	Poly – 3	RBF	Linear
	Correctly Predicted Subjects			
10	10	10	10	10
20	20	20	20	17
30	30	30	30	22

Without the seeds being part of the test data set, the percentage of correctly identified individuals is 65% for the first 10 individuals, 51.25 % for the first 20 individuals and 41.66% for the first 30 individuals. This is due to the increase in the subjects causes increase in the data and that there is a similarity among data points of different subjects.

It can be noticed that as the number of subjects increase the error increases across all classifiers. The polynomial kernel of order 2 performs better than the other kernels to identify individuals.

With the augmented test set containing the seed points, the percentage of correctly identified individuals is much higher compared to using the test set not containing the seed points. The percentage of correctly identified individuals is 100% for the first 10 individuals, 96.25% for the first 20 individuals and 93.33% for the first 30 individuals. The results of augmented test set is better than the test set not containing the seeds, because the synthetic data was generated using the seeds was used in building the models. All the models except of the linear SVM gives 100% accuracy in predicting the individuals. The linear SVM does better when there are fewer subjects to classify.

Tables 6-19 to 6-21 show the accuracy on target train data, target test data and outlier data and average accuracy on each of the folds based on maximum target training accuracy, maximum target test accuracy, maximum outlier accuracy and max average accuracy using Support Vector Data Descriptor. Average accuracy is the average accuracy of target train data and target test data. Nu is number of outliers in the data. Sigma is the width parameter of the Gaussian in the RBF kernel. The model is chosen based on the highest average accuracy. The parameters of the Gaussian kernel are chosen based on grid search on Nu and Sigma. The test data includes the seed data that generated the synthetic data. The models built here are the ones described in Figure 5-16.

Table 6-19 Accuracy of target prediction and outlier prediction for fold 1 on subject 1 using SVDD.

Sorted On	Target Train Data	Target Test Data	Outlier Data	Average Accuracy (Target Train + Target Test)	Nu	Sigma
maxTargetTrainAccuracy	96.19%	61.90%	74.67%	79.047%	0.07	8
maxTargetTestAccuracy	96.19%	61.90%	74.67%	79.047%	0.07	8
maxOutlierAccuracy	55.23%	0%	99.96%	27.61%	0.06	5.2
maxAverageAccuracy	96.19%	61.90%	74.67%	79.04%	0.07	8

Table 6-20 Accuracy of target prediction and outlier prediction for fold 2 on subject 1 using SVDD.

Sorted On	Target Train Data	Target Test Data	Outlier Data	Average Accuracy (Target Train + Target Test)	Nu	Sigma
maxTargetTrainAccuracy	94.28%	42.85%	86.32%	68.57%	0.16	6.4
maxTargetTestAccuracy	93.33%	42.85%	82.89%	68.09%	0.16	8
maxOutlierAccuracy	53.33%	0%	98.22%	26.66%	0.01	5
maxAverageAccuracy	94.28%	42.85%	86.32%	68.57%	0.16	6.4

Table 6-21 Accuracy of target prediction and outlier prediction for fold 3 on subject 1 using SVDD.

Sorted On	Target Train Data	Target Test Data	Outlier Data	Average Accuracy (Target Train + Target Test)	Nu	Sigma
maxTargetTrainAccuracy	99.03%	27.27%	92.45%	63.15%	0.04	7.9
maxTargetTestAccuracy	97.11%	59.09%	89.53%	78.10%	0.07	8
maxOutlierAccuracy	61.53%	0%	99.90%	30.76%	0.13	5
maxAverageAccuracy	97.11%	59.09%	89.53%	78.10%	0.07	8

Table 6-22 and 6-23 show prediction of subjects without the seed data included in the test set and with including the seeds in the test set respectively. Using the models and the approach described in Figure 5-17 we predict the subjects to whom the data belongs after validation. Without the seeds being part of the test data set, the percentage of correctly identified individuals is 40% for the first 10 individuals, 40% for the first 20 individuals and 31.03% for the first 29 individuals. The test results are not good because the bound includes the data belonging to other subjects. As a result there are higher misclassifications. It can be noticed that as the number of subjects increase the error increases.

With the seeds being part of the test data set, the percentage of correctly identified individuals is 60% for the first 10 individuals, 65% for the first 20 individuals and 44.82% for the first 29 individuals. The results are better because of the addition of the seeds into the test set that generated the synthetic data.

Table 6-22 Correctly predicted subjects on test data not containing seeds using SVDD.

No of Subjects	SVDD-RBF
	Correctly Predicted Subjects
10	4
20	8
29	9

Table 6-23 Correctly predicted subjects on test data containing seeds using SVDD.

No of Subjects	SVDD-RBF
	Correctly Predicted Subjects
10	6
20	13
29	13

Tables 6-24 to 6-26 show the accuracy on target train data, target test data and outlier data and average accuracy on each of the folds based on maximum target training accuracy, maximum target test accuracy, maximum outlier accuracy and max average accuracy using One Class Support Vector Machines. Average accuracy is the average accuracy of target train data and target test data. Nu is number of outliers in the data. Gamma is they hyper parameter of the RBF kernel. The model is chosen based on the highest average accuracy. The parameters of the Gaussian kernel are chosen based on grid search on Nu and Gamma. The test data includes the seed data that generated the synthetic data. The models built here are the ones described in Figure 5-16.

Table 6-24 Accuracy of target prediction and outlier prediction for fold 1 on subject 1 using OCSVM.

Sorted On	Target Train Data	Target Test Data	Outlier Data	Average Accuracy (Target Train + Target Test)	Nu	Gamma
maxTargetTrainAccuracy	98.100%	26.41%	71.43%	62.25%	0.09	0.02
maxTargetTestAccuracy	5.7100%	97.53%	9.52%	51.62%	0.06	0.01
maxOutlierAccuracy	98.100%	26.41%	71.43%	62.25%	0.09	0.02
maxAverageAccuracy	98.100%	26.41%	71.43%	62.25%	0.09	0.02

Table 6-25 Accuracy of target prediction and outlier prediction for fold 2 on subject 1 using OCSVM.

Sorted On	Target Train Data	Target Test Data	Outlier Data	Average Accuracy (Target Train + Target Test)	Nu	Gamma
maxTargetTrainAccuracy	81.90%	30.80%	66.67%	56.35%	0.20	0.02
maxTargetTestAccuracy	6.670%	88.17%	23.81%	47.42%	0.03	0.01
maxOutlierAccuracy	80%	27.49%	85.71%	53.75%	0.06	0.02
maxAverageAccuracy	76.19%	45.68%	61.90%	60.93%	0.19	0.01

Table 6-26 Accuracy of target prediction and outlier prediction for fold 3 on subject 1 using OCSVM.

Sorted On	Target Train Data	Target Test Data	Outlier Data	Average Accuracy (Target Train + Target Test)	Nu	Gamma
maxTargetTrainAccuracy	62.50%	58.81%	50%	60.66%	0.15	0.02
maxTargetTestAccuracy	14.42%	70.84%	27.27%	42.63%	0.15	0.02
maxOutlierAccuracy	39.42%	57.32%	54.55%	48.37%	0.20	0.02
maxAverageAccuracy	62.50%	58.81%	50%	60.66%	0.15	0.02

Table 6-27 and 6-28 show prediction of subjects without the seed data included in the test set and with including the seeds in the test set respectively. Using the models and the approach described in Figure 5-17 we predict the subjects to whom the data belongs after validation. Without the seeds being part of the test data set, the percentage of correctly identified individuals is 0% for the first 10 individuals, 0% for the first 20 individuals and 3.4% for the first 29 individuals. The test results are not good because the bound includes the data belonging to other subjects. One class SVM is a one sided bound in a nonlinear space, whereas SVDD is hypershpere in the nonlinear space. As a result there are higher misclassifications compared to SVDD. Only one subject could be identified correctly.

With the seeds being part of the test data set, the percentage of correctly identified individuals is 0% for the first 10 individuals, 0% for the first 20 individuals and 3.4% for the first 29 individuals. The results are the same compared to the test set without the seeds. This is due to the poor accuracy of the individual models to differentiate between the target data and the outlier data.

Table 6-27 Correctly predicted subjects on test data not containing seeds using OCSVM.

No of Subjects	OCSVM-RBF
	Correctly Predicted Subjects
10	0
20	0
29	1

Table 6-28 Correctly predicted subjects on test data containing seeds using OCSVM.

No of Subjects	OCSVM-RBF
	Correctly Predicted Subjects
10	0
20	0
29	1

6.5 Anomaly Detection

Using the models built above for normal walk detection, anomaly detection was carried out on abnormally slow walk data of the subject to detect if the abnormally slow walk would be predicted as anomaly. 3 models from above belonging to each individual are used to predict if the input slow stride from the subject is an anomaly. The accuracy of the 3 models is averaged to get the final accuracy. Table 6-29 and 6-30 show the average accuracy of models for an individual to predict slow walk data as not belonging to the subject using SVDD and OCSVM respectively. The SVDD does better than the OCSVM. The OCSVM does poorly because of the one-sided bound in the nonlinear space while the SVDD is a hypersphere in the nonlinear space. Also, the generalization of the OCSVM is bad and can be observed from Tables 6-24 to 6-26. The outlier detection accuracy is poor, there by misclassifying a lot of outliers to be target.

Table 6-29 Anomaly detection using SVDD

Subject	Accuracy
1	100%
2	100%
3	100%
4	100%
5	100%
6	100%
7	100%
8	100%
9	100%
10	93.33%
11	100%
12	100%
13	100%
14	100%
15	100%
16	100%
17	100%
18	100%
19	100%
20	75%
21	100%
22	100%
23	100%
24	100%
25	100%
26	93.33%
27	100%
28	100%
29	100%

Table 6-30 Anomaly detection using OCSVM

Subject	Accuracy
1	61.11%
2	91.67%
3	50%
4	77.78%
5	42.85%
6	66.67%
7	76.67%
8	72.22%
9	46.15%
10	60%
11	54.16%
12	44.44%
13	72.22%
14	71.428%
15	33.33%
16	45.833%
17	80%
18	100%
19	42.85%
20	66.67%
21	91.667%
22	40%
23	93.33%
24	16.667%
25	50%
26	66.67%
27	83.33%
28	86.667%
29	76.19%

Chapter 7

FUTURE WORK

Walk data was further classified into beginning of walk, mid walk and end of walk. It would be interesting to see if person identification or anomaly detection can be improved using only mid walk segments as start of walk and end of walk segments may not accurately represent one's walking pattern. This would require more data to be generated as we would have to leave out start of walk and end of walk segments.

In this thesis we generate synthetic data by interpolation as original data was insufficient and was resulting in overfitting. Synthetic data of 100 points was generated using two folds of data. Further experiments can be carried out by increasing the data size to see if increase in data could result in better results.

More walking data needs to be collected for each individual so that the need for synthetic data doesn't arise. Using synthetic data might give biased results. The models have to be validated again to obtain results on data not containing synthetic data. Person identification has to be carried out to see if individuals can be correctly predicted.

The parameters need to be tuned for the SVM to obtain optimal results. Basic grid search optimization is used to find parameters that best fit the model for the individual. Other techniques include grid search optimization with cross validation. This might help in tuning the model better. Other approaches to classification could be tried to see accuracy of the models can be improved. The width parameter of the Gaussian is limited to 8 as further increasing the width would allow outlier data to be classified as individual's data. The tradeoff between decrease in the outlier accuracy and increase in accuracy for target identification with increase in the width of the Gaussian of the RBF kernel has to be determined to set the limit of the width of the Gaussian while performing grid search.

The one class classifiers were used with RBF kernel, as this is the default choice. While polynomial kernel of order 2 resulted in better person identification using one versus rest classifiers, SVDD with polynomial kernel might give better results at identifying individual’s stride and hence aid in better person identification.

Currently, strides are randomly sampled to obtain training and testing data. It would be interesting to see if we could get the same person identification results if data used are sequences of strides for training and testing instead of randomly sampling strides.

Chapter 8

CONCLUSION

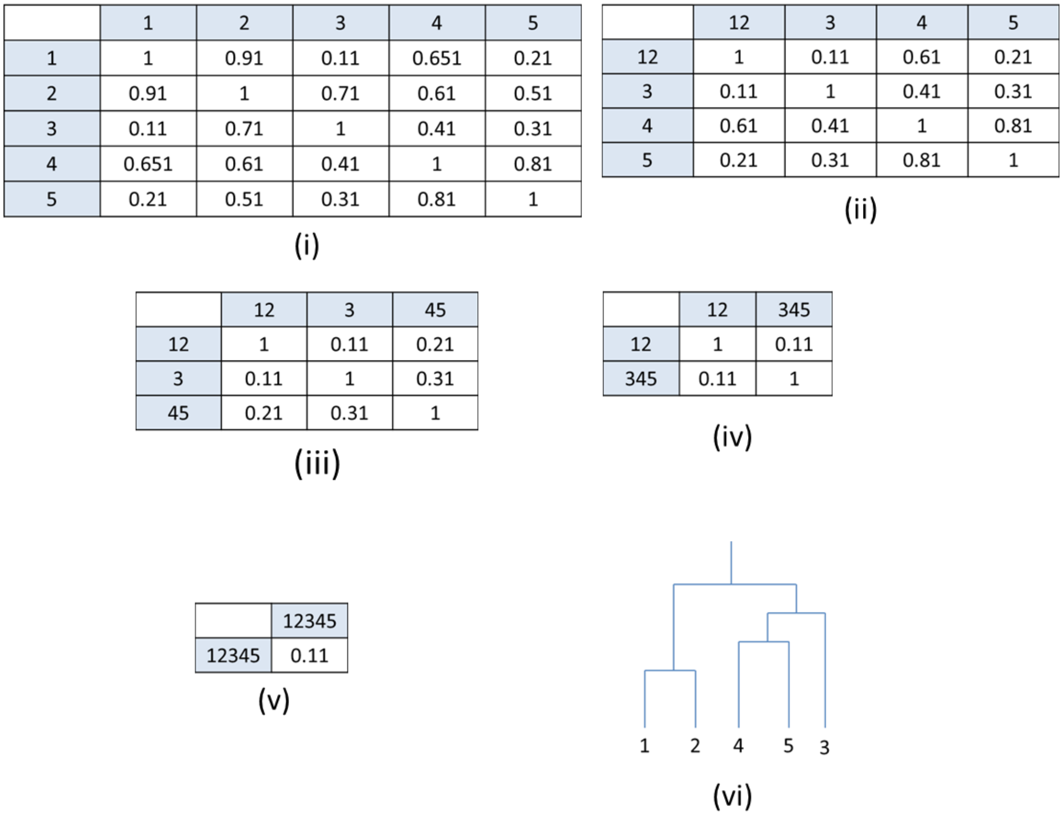
Using pressure monitoring floor we identify individuals walking on the floor and detect deviation in walking pattern to predict neurological problems on the onset so as to prevent it from becoming chronic. It can also be used for fall prediction in the elderly, which is one of the causes for death in the elderly. In this thesis, we managed to get high accuracy in classifying relevant walk segments from standing, short feet movement, turning and mixed segments. We also extracted gait parameters to build models for individuals for person identification and anomaly detection. We could identify 16 individuals out of 30 individuals on validation data not containing seed data points which was used in synthetic data generation for training the models. We could identify all the individuals on including the seed data on validation data which included seed data points. Using SVDD, we could identify abnormally slow walking data as anomalous with very high accuracy. The results of this research are promising for person identification and anomaly detection.

Appendix A

WORKING OF ALGORITHMS BY EXAMPLE

A.1 Hierarchical Clustering using Complete Linkage

Figure 1 shows merging of 5 points, (vi) shows a dendrogram which is a tree like structure that shows the merging between the clusters. [2].



	1	2	3	4	5
1	1	0.91	0.11	0.651	0.21
2	0.91	1	0.71	0.61	0.51
3	0.11	0.71	1	0.41	0.31
4	0.651	0.61	0.41	1	0.81
5	0.21	0.51	0.31	0.81	1

(i)

	12	3	4	5
12	1	0.11	0.61	0.21
3	0.11	1	0.41	0.31
4	0.61	0.41	1	0.81
5	0.21	0.31	0.81	1

(ii)

	12	3	45
12	1	0.11	0.21
3	0.11	1	0.31
45	0.21	0.31	1

(iii)

	12	345
12	1	0.11
345	0.11	1

(iv)

	12345
12345	0.11

(v)

(vi)

Figure 1 is a similarity matrix of 5 points clustered hierarchically using complete linkage. (i) is the similarity matrix, (ii)-(v) is the similarity matrix after merging at each stage. (vi) is the dendrogram of the clustering.

Comparison between Single Linkage and Complete Linkage

Figure 2 shows a set of points that are clustered using single linkage and complete linkage criterion. The dendrogram of the clustering of single linkage and complete linkage of the data shown in figure 2 is shown in figure 3. On both the clustering the dendrogram is cut at a certain height to obtain two clusters. The clusters of single linkage are defined by proximity while the clusters of the complete linkage are defined by global closeness [2]. Form figure 3, it can be observed that the points within the cluster are closer to each other using complete linkage when compared to single linkage.

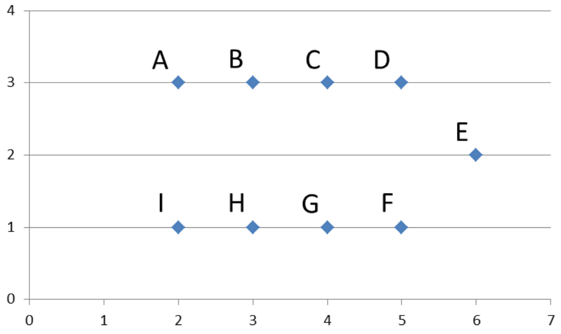


Figure 2 Set of points to be clustered [2].

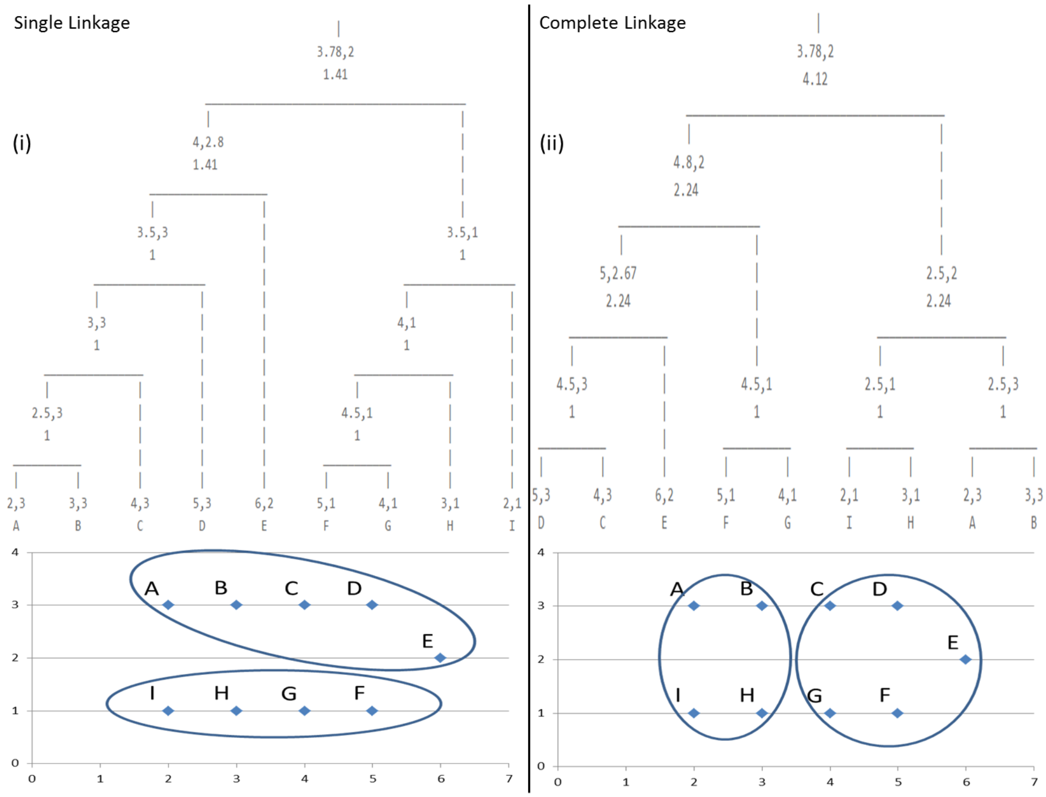
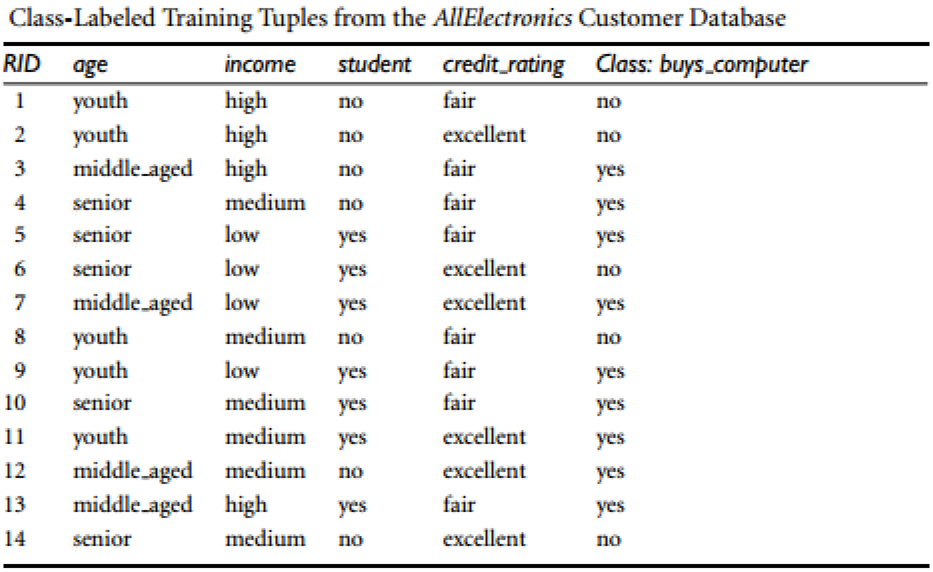


Figure 3 Clustering using single linkage and complete linkage. It also shows the clusters obtained when the dendrogram is cut to give 2 clusters for single linkage and complete linkage. [2][8]

A.2 Working of CART Algorithm by Example

The working of the CART algorithm is explained below using an example data set shown in figure 4. The CART algorithm creates a binary split using Gini Index. The attribute with the lowest Gini Index is used as the splitting attribute. If the attribute have more than 2 values, the split is created using one value versus the rest of values. This process is repeated for all the values of the attribute. The example shows an exhaustive search to find the attribute to split by at the root level using a binary split. [2].



Class-Labeled Training Tuples from the *AllElectronics* Customer Database

RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Figure 4 Sample data set [2]

$$Gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

	Splitting Attribute:: Age		
	Youth	Middle-Aged	Senior
Yes	2	4	3
No	3	0	2

	Splitting Attribute:: Age	
	Youth & Senior	Middle-Aged
Yes	5	4
No	5	0

Gini index for the subsets {youth} and {middle-aged, senior} is given by:

$$= \frac{10}{14}\left(1 - \left(\frac{5}{10}\right)^2 - \left(\frac{5}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{4}{4}\right)^2 - \left(\frac{0}{4}\right)^2\right)$$

= 0.3571

	Splitting Attribute:: Age	
	Youth & Middle - Aged	Senior
Yes	6	3
No	3	2

Gini index for the subsets {youth, middle-aged} and {senior} is given by:

$$= \frac{9}{14}\left(1 - \left(\frac{6}{9}\right)^2 - \left(\frac{3}{9}\right)^2\right) + \frac{5}{14}\left(1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2\right)$$

= 0.4571

	Splitting Attribute:: Age	
	Youth	Middle-Aged & Senior
Yes	2	7
No	3	2

Gini index for the subsets {youth} and {middle-aged, senior} is given by:

$$= \frac{5}{14}\left(1 - \left(\frac{2}{5}\right)^2 - \left(\frac{3}{5}\right)^2\right) + \frac{9}{14}\left(1 - \left(\frac{7}{9}\right)^2 - \left(\frac{2}{9}\right)^2\right)$$

= 0.3936

	Splitting Attribute:: Income		
	High	Low	Medium
Yes	2	3	4
No	2	1	2

	Splitting Attribute:: Income	
	High	Low &Medium
Yes	2	7
No	2	3

Gini index for the subsets {medium, low} and {high} is given by:

$$= \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right) + \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right)$$

= 0.4428

	Splitting Attribute:: Income	
	High & Medium	Low
Yes	6	3
No	4	1

Gini index for the subsets {high, medium} and {low} is given by:

$$= \frac{10}{14}\left(1 - \left(\frac{6}{10}\right)^2 - \left(\frac{4}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2\right)$$

= 0.4499

	Splitting Attribute:: Income	
	High & Low	Medium
Yes	5	4
No	3	2

Gini index for the subsets {high, low} and {medium} is given by:

$$= \frac{8}{14}\left(1-\left(\frac{5}{8}\right)^2-\left(\frac{3}{8}\right)^2\right)+\frac{6}{14}\left(1-\left(\frac{4}{6}\right)^2-\left(\frac{2}{6}\right)^2\right)$$

= 0.4582

	Splitting Attribute:: Credit Rating	
	Fair	Excellent
Yes	6	3
No	2	3

Gini index for credit rating is given by:

$$= \frac{8}{14}\left(1-\left(\frac{6}{8}\right)^2-\left(\frac{2}{8}\right)^2\right)+\frac{6}{14}\left(1-\left(\frac{3}{6}\right)^2-\left(\frac{3}{6}\right)^2\right)$$

= 0.4285

	Splitting Attribute:: Student	
	Yes	No
Yes	6	3
No	1	4

Gini index for student is given by:

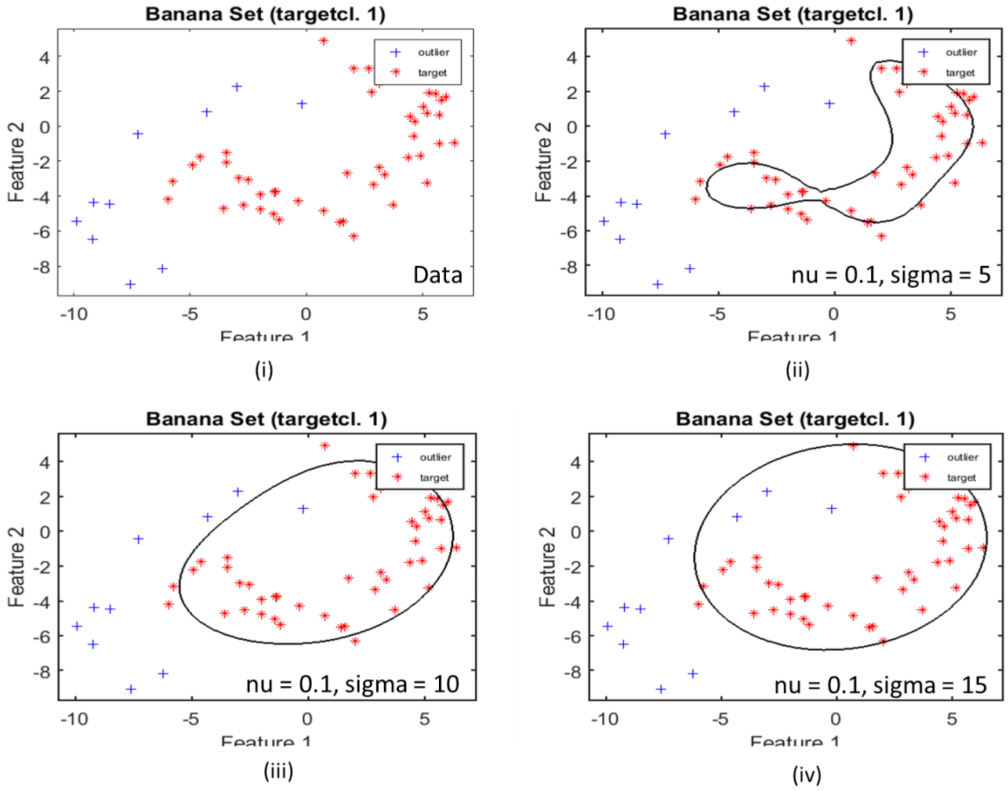
$$= \frac{7}{14}\left(1-\left(\frac{6}{7}\right)^2-\left(\frac{1}{7}\right)^2\right)+\frac{7}{14}\left(1-\left(\frac{3}{7}\right)^2-\left(\frac{4}{7}\right)^2\right)$$

= 0.3775

The binary split by age {Youth & Senior} (or {Middle-Aged}) gives the highest reduction in impurity, which is equal to 0.459 – 0.3571 = 0.1019.

A.3 Formation of decision boundary using support vector data descriptor on banana data set

Using the dd toolbox provided by Tax, et al [9], a target data set in the shape of a banana and the outlier data set outside the banana data set are generated. The dd toolbox [9] is used to learn a decision boundary around the target data using radial basis function kernel for different widths of the Gaussian and by setting the percentage of outliers in the target data to be equal to 10 percent. Figure 5 shows the data and the decision boundary around the target data with increasing *sigma* (width of the Gaussian) and *nu* = 0.1 (fraction of outliers in the target data).



(i)

(ii)

(iii)

(iv)

Figure 5 SVDD boundary (i) Target data is red and outlier data is blue. (ii) Decision boundary around the target data when sigma = 5. (iii) Decision boundary around the target data when sigma = 10. (iv) Decision boundary around the target data when sigma = 15. [9]

REFERENCES

[1] Perry, J., & Burnfield, J. M. (2010). Gait analysis: normal and pathological function. Thorofare: Slack.

[2] Han, J., Kamber, M., & Pei, J. (2012). Data mining: concepts and techniques. Waltham, MA: Morgan Kaufmann.

[3] Russell, S. J., Norvig, P., & Davis, E. (2010). Artificial intelligence: a modern approach. Upper Saddle River, NJ: Prentice Hall.

[4] Abu-Mostafa, Y. S., Magdon-Ismail, M., & Lin, H. (2012). Learning from data: a short course. S. l.: AML.

[5] Burges, Chris J.C. "A Tutorial on Support Vector Machines for Pattern Recognition." Microsoft Research. N.p., 01 Jan. 1998. Web.

[6] Tax, David M.j., and Robert P.w. Duin. "Support Vector Data Description." Machine Learning 54.1 (2004): 45-66. Web.

[7] Schölkopf, Bernhard, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. "Estimating the Support of a High-Dimensional Distribution." Neural Computation 13.7 (2001): 1443-471. Web.

[8] Teknomo, Kardi. Online Hierarchical Clustering Calculator. N.p., n.d. Web. <http://people.revoledu.com/kardi/tutorial/Clustering/Online-Hierarchical-Clustering.html>.

[9] Tax, and D.M.J. "Dd_tools." DDtools, the Data Description Toolbox for Matlab. N.p., June 2015. Web. <http://prlab.tudelft.nl/david-tax/dd_tools.html>.

[10] Oluwadare, O. (2015). Gait analysis on a smart floor for health monitoring. Arlington, TX: University of Texas at Arlington.

[11] Data Mining Algorithms In R/Clustering/Hybrid Hierarchical Clustering. (n.d.).Retrieved from https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/Hybrid_Hierarchical_Clustering

[12] A Practical Guide to Support Vector Classification. (n.d.). Retrieved from http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[13] Bracewell, R. N. (2000). The Fourier transform and its applications: solutions manual. New York: : McGraw-Hill.

[14] (n.d.). Retrieved from http://my.me.queensu.ca/People/Deluzio/Gait.html

[15] Whittle, M. W. (2003). Gait analysis: An Introduction. London: Butterworth-Heinemann Ltd.

[16] Murray MP, Sepic SB, Gardner GM et al. (1978) Walking patterns of men with parkinsonism.American Journal of Physical Medicine 57: 278–294.

[17] Robinson JL, Smidt GL. (1981) Quantitative gait evaluation in the clinic. Physical Therapy 61:351–353.

[18] Rose GK. (1983) Clinical gait assessment: a personal view. Journal of Medical Engineering and Technology 7:273–279.

[19] Verghese, J., Lipton, R. B., Hall, C. B., Kuslansky, G., Katz, M. J., & Buschke, H. (2002). Abnormality of Gait as a Predictor of Non-Alzheimer's Dementia. New England Journal of Medicine, 347(22), 1761-1768. doi:10.1056/nejmoa020441

[20] Sudarsky L. Geriatrics: gait disorders in the elderly. N Engl J Med 1990;322:1441-1446

[21] Fuh J-L, Lin K-N, Wang S-J, Ju T-H, Chang R, Liu H-C. Neurologic diseases presenting with gait impairment in the elderly. J Geriatr Psychiatry Neurol 1994;7:89-92

[22] Rosen WG, Terry RD, Fuld PA, Katzman R, Peck A. Pathological verification of ischemic score in the differentiation of dementias. Ann Neurol 1980;7:486-488

[23] Whitman GT, Tang Y, Lin A, Baloh RW, Tang T. A prospective study of cerebral white matter abnormalities in older people with gait dysfunction. Neurology 2001;57:990-994[Erratum, Neurology 2001;57:1942.]

[24] C. BenAbdelkader, R. Cutler and L. Davis, "Stride and cadence as a biometric in automatic person identification and verification," Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition, Washington, DC, USA, 2002, pp. 372-377. doi: 10.1109/AFGR.2002.1004182

[25] Liang Wang, Tieniu Tan, Huazhong Ning and Weiming Hu, "Silhouette analysis-based gait recognition for human identification," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 12, pp. 1505-1518, Dec. 2003. doi: 10.1109/TPAMI.2003.1251144

[26] C. BenAbdelkader, R. Cutler and L. Davis, "Person identification using automatic height and stride estimation," Object recognition supported by user interaction for service robots, 2002, pp. 377-380 vol.4. doi: 10.1109/ICPR.2002.1047474

[27] Zhang, Z., & Troje, N. F. (2005). View-independent person identification from human gait. Neurocomputing, 69(1-3), 250-256. doi:10.1016/j.neucom.2005.06.002

[28] Zhang, T., Wang, J., Xu, L., & Liu, P. (n.d.). Fall Detection by Wearable Sensor and One-Class SVM Algorithm. Intelligent Computing in Signal Processing and Pattern Recognition Lecture Notes in Control and Information Sciences, 858-863. doi:10.1007/11816515_104

[29] Larry M. Manevitz,Malik Yousef One-Class SVMs for Document Classification Journal of Machine Learning Research 2 (2001) 139-154\

[30] Zhou, J., Chan, K., Chong, V., & Krishnan, S. (2005). Extraction of Brain Tumor from MR Images Using One-Class Support Vector Machine. 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference. doi:10.1109/iembs.2005.1615965

[31] Zhao Y., Li B., Li X., Liu W., Ren S. (2005) Customer Churn Prediction Using Improved One-Class Support Vector Machine. In: Li X., Wang S., Dong Z.Y. (eds) Advanced Data Mining and Applications. ADMA 2005. Lecture Notes in Computer Science, vol 3584. Springer, Berlin, Heidelberg

[32] Cola, G., Avvenuti, M., Vecchio, A., Yang, G., & Lo, B. (2015). An On-Node Processing Approach for Anomaly Detection in Gait. IEEE Sensors Journal, 15(11), 6640-6649. doi:10.1109/jsen.2015.2464774

[33] Begg, R., & Palaniswami, M. (n.d.). Recognition of Gait Patterns Using Support Vector Machines. Computational Intelligence for Movement Sciences. doi:10.4018/9781591408369.ch008

[34] Martinez, M. T., Leon, P. L., & Keeley, D. (2016, February 01). Anomaly Detection for Prediciting Falls Risk using Smartphone Gait Data. Retrieved from https://www.osti.gov/scitech/servlets/purl/1344828

[35] Nguyen, T., Huynh, H., & Meunier, J. (2016, October 26). Skeleton-Based Abnormal Gait Detection. Retrieved from http://mdpi.com/1424-8220/16/11/1792

BIOGRAPHICAL INFORMATION

Suhas Mandikal Rama Krishna Reddy completed his Bachelor’s degree in Information Science and Engineering from Visvesvaraya Technological University in Belgaum, Karnataka, India in 2012. He started his Master’s Degree in Computer Science at University of Texas at Arlington in 2014 specializing in Artificial Intelligence. His interests include Statistics, Machine Learning Theory and Deep Learning.