Human Activity Recognition through Smartphone Sensor Data

by

Abhishek Bagade- 163059005

Jitendra Bafna - 163050015

Rahul Anand- 163050084

Sudhanshu Deshmukh- 163050050

Department of Computer Science
IIT Bombay

April 2017

Abstract

Human activity recognition (HAR) is a very important problem which has gained prominence due to current advent of activity trackers like fitbit,samsung fit etc. The use of specialized hardware can be easily avoided by using the inbuilt sensors in smartphones like Accelerometer, gyroscope and compass. The sensors have high sensitivity and predict the activity fairly accurately. The dataset we are using is available publicly and contains 561 features which have been made by discretizing the continuous values over a fixed time interval, the data has been pre-processed and is normalized. The problem to be solved is a classical classification problem and we have studied and compared the results of various classification algorithms. We aim to find the classifier which provides the best result.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The recent advances in development of mobile computing and mobile devices has increased the computing as well as the sensing powers available to a common user .One aspect that has benefitted most of the users is the possiblity of continuous sensing [1].

This is an active research area with the main purpose of extracting knowledge from the data acquired by pervasive sensors. Particularly, the recognition of human activities has become a task of high interest within the field, especially for medical, military, and security applications.

The problem of HAR has been tackled by employing a number of classifiers like Support Vector Classifier, Gaussian Discriminant Analysis, Neural Networks, K-Means classifier, Hidden Markov Models etc [2] [1] [3] [4].

1.1 General structure of a HAR

Similar to other machine learning classification applications, HAR requires two stages, i.e., training and testing. Figure 1.1 illustrates the common phases involved in these two processes. The training stage initially requires a time series dataset of measured attributes from individuals performing each activity. The time series are split into time windows to apply feature extraction thereby filtering relevant information in the raw signals. Later, learning methods are used to generate an activity recognition model from the dataset of extracted features.

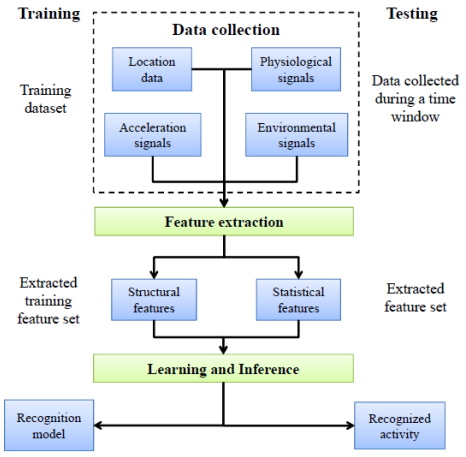

FIGURE 1.1: General data flow for training and testing in HAR

[5] expressed a generic data acquisition architecture for HAR systems, as shown in Figure 1.2. In the first place, wearable sensors are attached to the persons body to measure attributes of interest such as motion, location, temperature, ECG, among others. These sensors should communicate with an integration device (ID), which can be a cellphone, a PDA, a laptop or a customized embedded system. The main purpose of the ID is to preprocess the data received from the sensors and, send them to an application server for real time monitoring, visualization, and analysis.

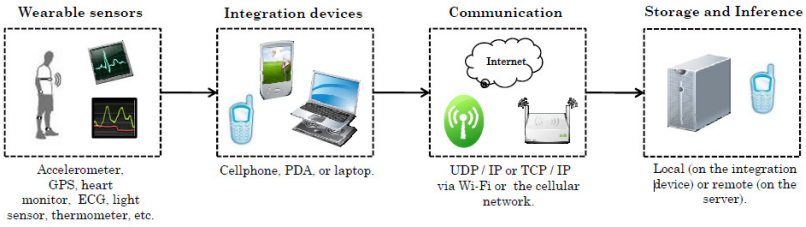

FIGURE 1.2: Generic data acquisition system for HAR

1.2 Challenges in HAR

Though there have been considerable advancements in HAR domain, there are some problems that severely restricts the real-time usage. Some of these challenges are -

1. Selection of attributes/sensors for measurement,
2. feature extraction from continuous data.

3. The construction of a portable, unobtrusive, and inexpensive data acquisition system. Though this is mitigated in large part due to advancements in mobile technology.
4. the design of feature extraction and inference methods.
5. reducing the computational complexity of the problem.
6. making the model platform agnostic and compatible with all types of smart phones.
7. the implementation in mobile devices meeting energy and processing requirements

1.3 Objectives of study

With this background, we therefore set the objectives for our current study as follows -

1. To study and formulate a HAR problem.
2. To study the effectiveness of classification algorithms on HAR dataset
3. To find out the challenges and possible solutions to HAR dataset problems.

Chapter 2

HAR Dataset

The Dataset taken for this study is a publicly available courtesy Anguita et al . The below table shows the specifications of the dataset. 2.1.

Some comments -

1. Accelerometer gives acceleration signal and Gyroscope gives velocity signals.The accelerometer helps to detect 3D motion whereas gyroscope gives 2d information
2. Acceleration signal can be broken down through signal processing into two signals gravitational signal (low frequency) and body signal.
3. The 17 signals obtained from the raw data are mentioned as below. Eg. Body acc signal has been taken in time domain as well as frequency domain. Eg. Gravity Acc signal has been taken only in time domain.

Table 2.1: Description of HAR dataset

Parameter	—Description
No of users	—30
Raw Data	—acceleration XYZ (time); velocity XYZ (time) 50 Hz
Raw Data (Given with the dataset)	—Total acceleration XYZ (observed & sampled 7352x128); Velocity XYZ (observed & sampled 7352x128); Body acceleration XYZ (calculated & sampled 7352x128)
Processed Signals	—17
No of features extracted	—561
No of labels	—6
No of train instances	—7352
No of test instances	—2947

4. Each of these 17 processed signal (continuous) have been sampled for every 2.56 seconds of the signal. Hence, each time window contains 128 rows (2.56 seconds * 50 Hz).

5. After sampling the 17 processed signals at a 50% overlapping window of 2.56 seconds and combining the observation from different users, we have 7352 instances with 128 values of each signal (7352x128).

6. 8 of these 17 signals have XYZ components. Hence, in total we can say that we have $(17-8)*1+8*3 = 33$ signals. For each signal, 17 parameters (such as mean, std, min, max etc) are calculated. Hence, for each time window of the 33 signals, we have $33 * 17 = 561$ feature vectors. (7352x561)

The activity labels for prediction are specified in the Table 2.2.

TABLE 2.2: Prediction activity Labels

Activity	Label
Walking	1
Walking Upstairs	2
Walking downstairs	3
Sitting	4
Standing	5
Laying	6

Chapter 3

Gradient Boosting Classifier

3.1 Summary of Gradient Boosting :

Boosting is a machine learning algorithm for primarily reducing bias, and also variance in supervised learning, and a family of machine learning algorithms which convert weak learners to strong ones.

3.2 Implementation Details of Gradient Boosting :

We used boosting classifier in the form of Gradient boosting classifier provided by sklearn library. The overall accuracy of the classifier was 93.5 percent which was okay but not accurate enough as other models. The gradient boosting classifer works similar to AdaBoost by setting appropriate parameters. The accuracy was not upto the mark so we didn't experiment more with it, the principle by which it works was very interesting and our guess was that it didn't work due to the very low variance in our data.

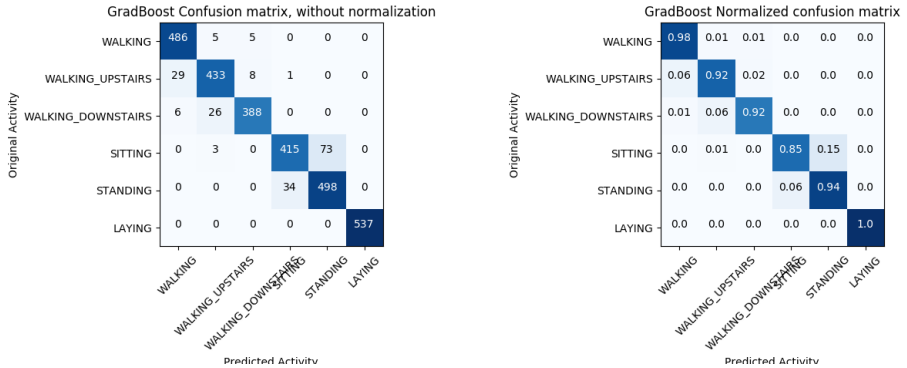

FIGURE 3.1: Confusion Matrix of Gradient Boosting

Chapter 4

Bagging Classifier using SVM Linear kernel

4.1 Summary of Bagging Classifier :

Bagging, is a machine learning meta-classifier designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It reduces variance and helps to avoid over fitting. Although it is usually applied to decision tree methods, it can be used with any type of method. Bagging is a special case of the model averaging approach. Bagging picks up a random data points with replacement to decrease the variance of the data.

4.2 Implementation Details of Bagging Classifier :

We implemented the default implementation of bagging classifier which uses Decision trees, this approach produced accuracy better than the vanilla Decision tree classifier but as the decision tree classifier didn't produce results with high precision the overall precision or accuracy was still very less. We then used bagging classifier over SVC with a linear kernel as it was the model giving the highest accuracy, the overall accuracy should've increased due to bagging classifier, but it was still less than the basic SVM classifier with linear kernel.

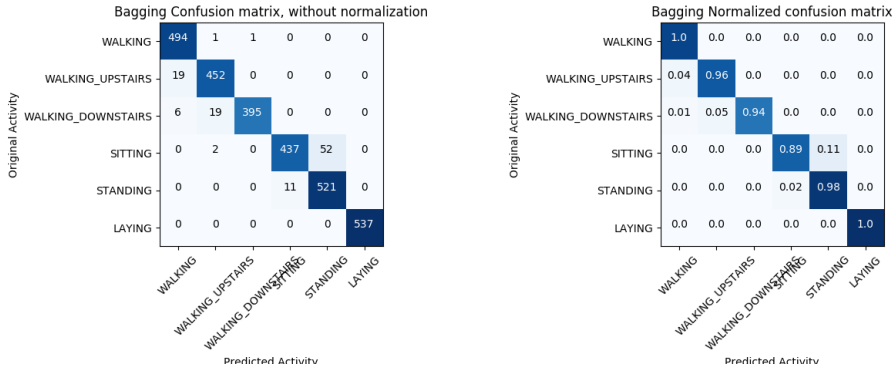

FIGURE 4.1: Confusion Matrix of Bagging Classifier using SVM Linear kernel

Chapter 5

Support Vector Classification for HAR

5.1 SVC summary

A support vector machine constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

Some advantages of support vector machine include effectiveness for high dimensional spaces,uses subset of training points in decision function (called support vectors) and different Kernel functions can be specified for the decision function. Some disadvantages are they do not provide probabilistic estimates.

SVC, NuSVC and LinearSVC are classes capable of performing multi-class classification on a dataset.SVC and NuSVC implement the one-against-one approach (Knerr et al., 1990) for multi- class classification.On the other hand, LinearSVC implements one-vs-the-rest multi-class strategy.

Given training vectors $x_i \in \mathbb{R}^p$,$i = 1, ..., n$, in two classes, and a vector $y \in \{1, -1\}^n$, SVC solves the following primal problem:

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i \tag{5.1}$$

$$\text{subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i,$$

$$\zeta_i \geq 0,$$

$$i = 1, ..., n$$

The decision function is:

$$f(x) = \left(\sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + \rho \right) \tag{5.2}$$

5.2 Implementation of SVC

For the problem at hand, SVC was implemented with different kernels namely RBF, Linear and Polynomial. The input consisted of the entire 7352 instances with 500 features each. We managed to get maximum accuracy from the linear kernel which performed best in all classification methods with 96.43% . The confusion matrices are also produced for reference in Figures below:

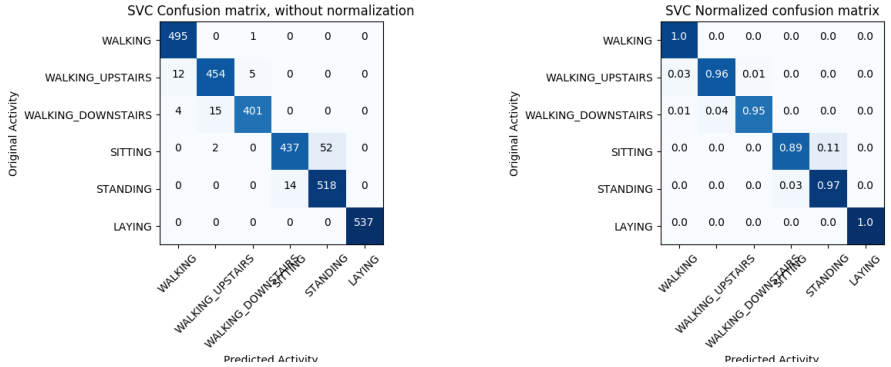

FIGURE 5.1: Confusion Matrix of Support Vector Classification

Chapter 6

k-nearest neighbors algorithm

6.1 KNN summary :

In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression.The input consists of the k closest training examples in the feature space.

6.2 Implementation Details of K-nearest neighbors:

KNN as a classifier suffers from the 'curse of dimensionality'. As the no of features increases the volume of the space spanned by the feature space increase and hence the calculation of K-nearest-neighbours becomes harder. As expected with our 561 dimensional feature space KNN performed abysmal, though the results were not terribly bad but the computing time used and the accuracy obtained immediately made the classifier unfavourable.

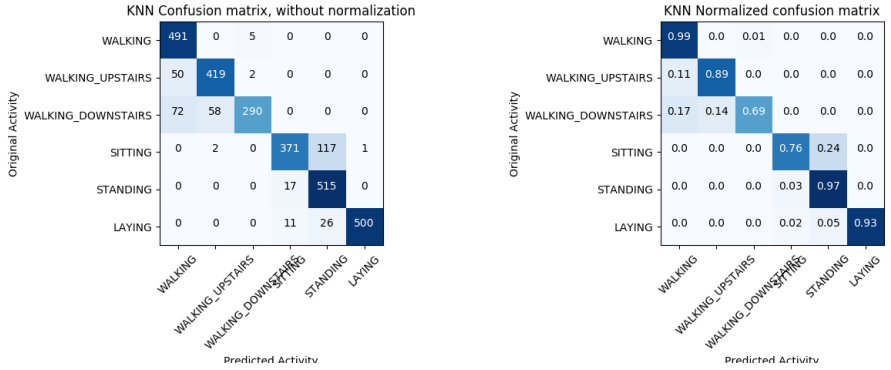

FIGURE 6.1: Confusion Matrix of K-nearest neighbors

Chapter 7

Random Forest

7.1 Summary of RF :

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set

7.2 Implementation Details of Random Forest :

We used Random forests with 100 estimators, the accuracy though good was not comparable to that of other models. We experimented with various values of estimators and found the best results for value of 100.

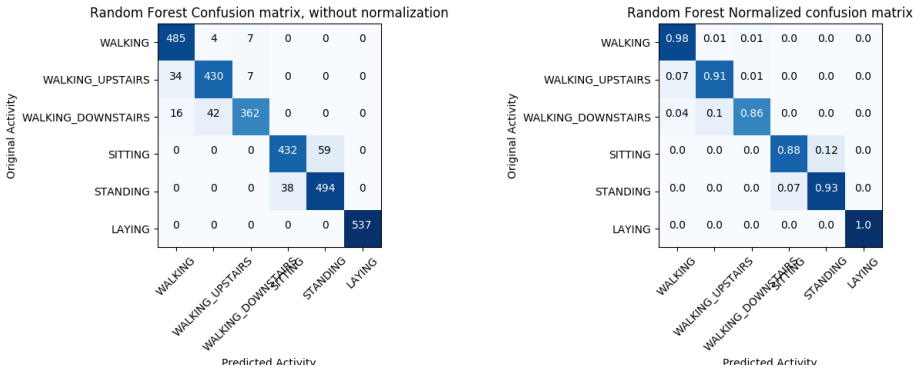

FIGURE 7.1: Confusion Matrix of Random Forest

Chapter 8

Principal component analysis

8.1 PCA Summary :

Principal components analysis is a procedure for identifying a smaller number of uncorrelated variables, called "principal components", from a large set of data. The goal of principal components analysis is to explain the maximum amount of variance with the fewest number of principal components. Principal components analysis is commonly used in the social sciences, market research, and other industries that use large data sets. Principal components analysis is commonly used as one step in a series of analyses. You can use principal components analysis to reduce the number of variables and avoid multicollinearity, or when you have too many predictors relative to the number of observations.

8.2 Implementation Details of PCA

We used PCA for plotting the values in a 2-D graph. The graph helped us visualize the dataset better and form an intuitive understanding of the dataset and the classifiers to be used. PCA helps to sompress the feature space in n dimensions by setting n_components parameter

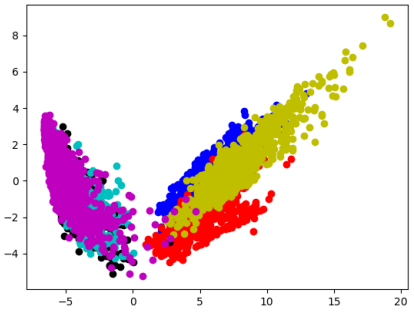

FIGURE 8.1: Principal component analysis

Chapter 9

Quadratic Discriminant Analysis

9.1 QDA Summary :

Quadratic discriminant analysis (QDA) is closely related to linear discriminant analysis (LDA), where it is assumed that the measurements from each class are normally distributed. Unlike LDA however, in QDA there is no assumption that the covariance of each of the classes is identical. When the normality assumption is true, the best possible test for the hypothesis that a given measurement is from a given class is the likelihood ratio test.

9.2 Implementation Details of QDA :

Compared to LDA the accuracy for QDA was very less. This maybe due to the fact QDA leads to quadratic decision surfaces.

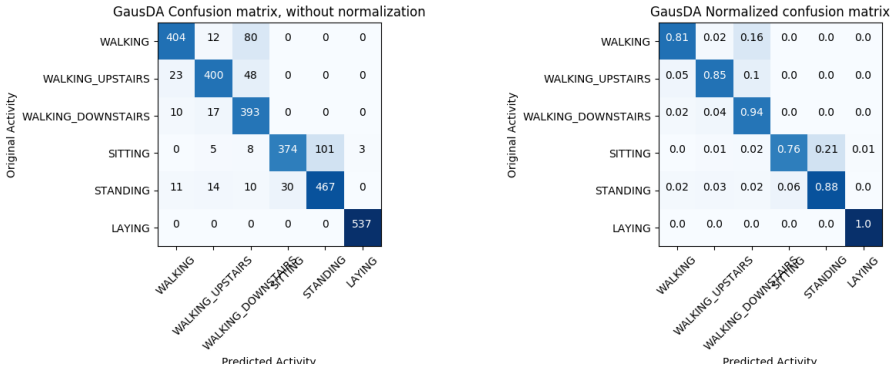

FIGURE 9.1: Confusion Matrix of Quadratic Discriminant Analysis

Chapter 10

Linear discriminant analysis

10.1 LDA Summary :

Linear Discriminant Analysis (LDA) is most commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications. The goal is to project a dataset onto a lower-dimensional space with good class-separability in order avoid overfitting (curse of dimensionality) and also reduce computational costs. The general LDA approach is very similar to a Principal Component Analysis, but in addition to finding the component axes that maximize the variance of our data (PCA), we are additionally interested in the axes that maximize the separation between multiple classes (LDA).

In general, dimensionality reduction does not only help reducing computational costs for a given classification task, but it can also be helpful to avoid over fitting by minimizing the error in parameter estimation (curse of dimensionality).

10.2 Implementation Details of LDA :

LDA returned the second highest accuracy for our model when used as classifier which was expected due the dataset being more favourable to 'simpler' classifiers.the classifier fits a Gaussian distribution to every class and then for every point generates probability for each class and selects the class with maximum probability.

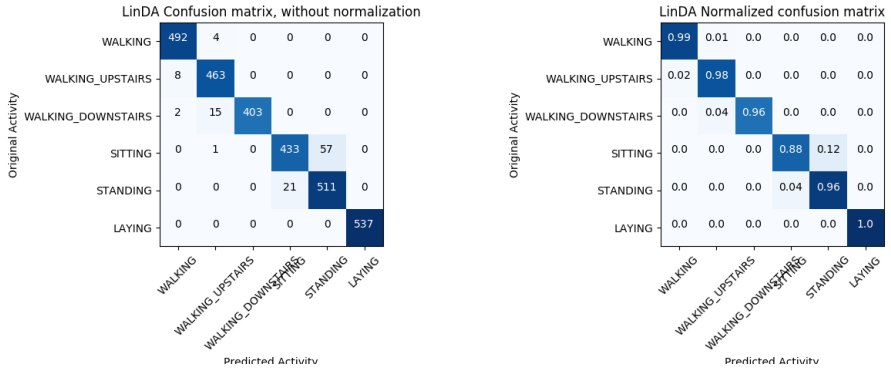

FIGURE 10.1: Confusion Matrix of Linear discriminant analysis

Chapter 11

Neural Network (NN) Classification for HAR

11.1 NN summary

A neural network is a powerful computational data model that is able to capture and represent complex input/output relationships. The motivation for the development of neural network technology stemmed from the desire to develop an artificial system that could perform "intelligent" tasks similar to those performed by the human brain. Neural networks resemble the human brain in the following two ways:

1. A neural network acquires knowledge through learning.
2. A neural network's knowledge is stored within inter-neuron connection strengths known as synaptic weights.

The true power and advantage of neural networks lies in their ability to represent both linear and non-linear relationships and in their ability to learn these relationships directly from the data being modelled Traditional linear models are simply inadequate when it comes to modelling data that contains non-linear characteristics.

11.2 Implementation of Neural Network for HAR

The NN used to classify is the Multilayer Perceptron Classifier as illustrated in Figure 11.1.

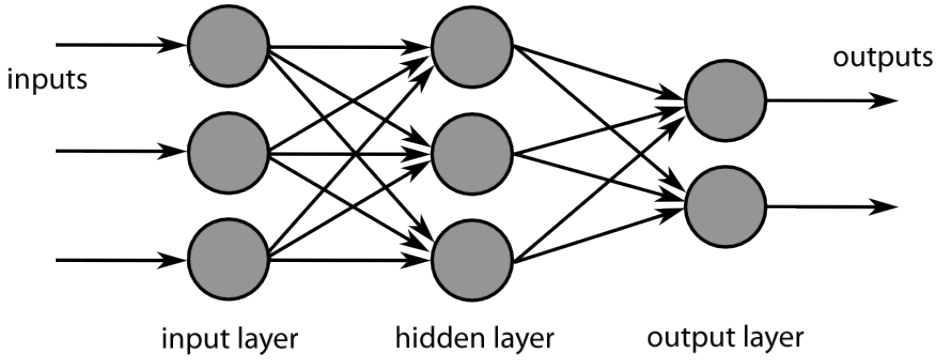

FIGURE 11.1: Architecture of MLP

The diagram above is an two hidden layer Multilayer Perceptron (MLP). The inputs are fed into the input layer and get multiplied by interconnection weights as they are passed from the input layer to the first hidden layer. Within the first hidden layer, they get summed then processed by a nonlinear function (RELU is used here). As the processed data leaves the first hidden layer, again it gets multiplied by interconnection weights, then summed and processed by the second hidden layer. Finally the data is multiplied by interconnection weights then processed one last time within the output layer to produce the neural network output.

The neural network learns through back-propagation algorithm. In this algorithm we ensure that the network learns and corrects its weight according to the difference in predicted and actual value. this learning is transmitted in reverse direction hence 'back' propagation

The MLP classifier tries to minimize cost after each back-propagation through a optimizer. The "Adam" optimizer produced sufficiently good results(95% accuracy) as compared to Stochastic Gradient Descent(79%). The layer configuration used is 80 hidden layers since the features are high while Dataset is small.

The results of the classifier was 94.60% accuracy which was good but was worse than that producedby sipler models like SVM. This suggested that the data was definitely being overfitted in neural network. The detailed analysis is given below.

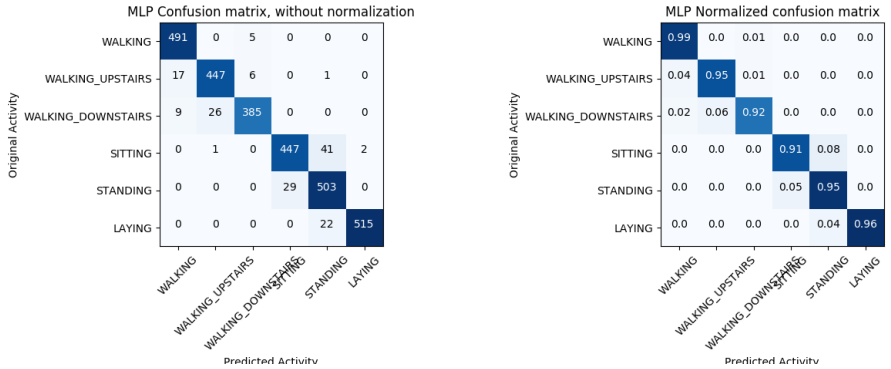

FIGURE 11.2: Confusion Matrix of Neural Network

Chapter 12

Discussion on Results

12.1 Feature Engineering

The major component of feature engineering was generating discrete features from continuous data. This task was handled by the publisher of dataset and reading about it in the paper published we got to know about the techniques used to analyse continuous data and generate discrete data from sensors. After elementary analysis of the data the activity labels can again be divided in two main types - Static and Dynamic. The static activities involve minimum movement and hence the readings of the sensors don't vary much.It includes Sitting (4), Standing (5) and Laying (6). The dynamic activities are Walking (1), Walking upstairs (2) and Walking downstairs (3).

12.1.1 Principal component analysis

We performed PCA (Principal component analysis) to analyze the data set and get a general idea of the distribution of data points. PCA does unsupervised dimensional

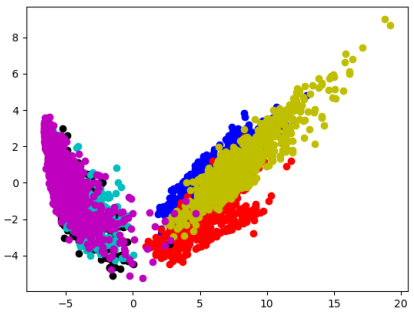

FIGURE 12.1: Scatter plot of activities with first two principal components

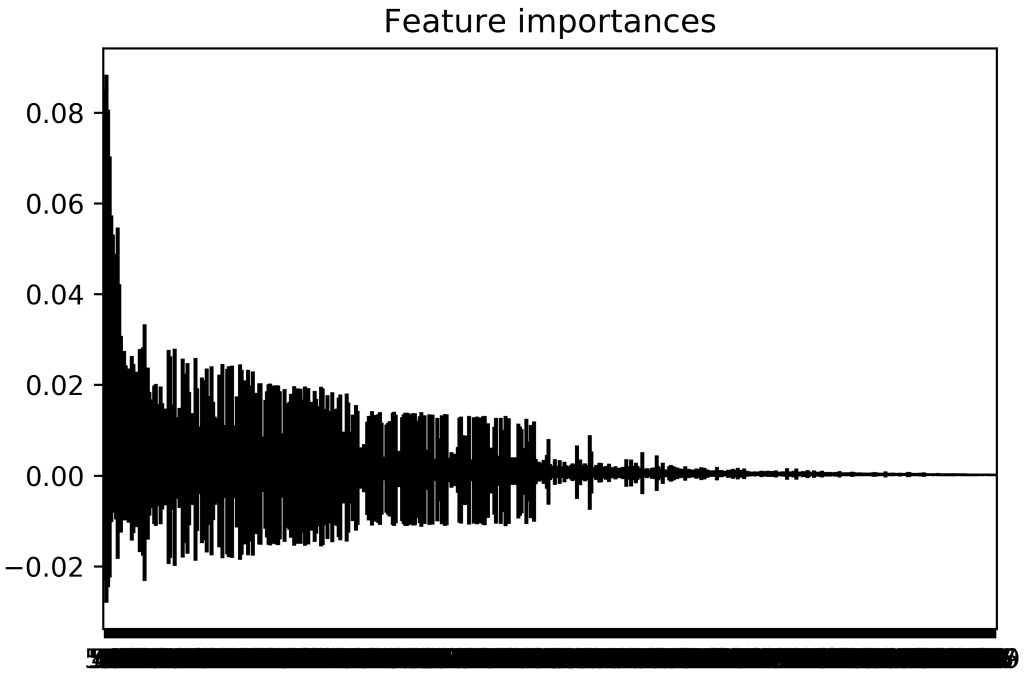

FIGURE 12.2: Feature importance

reduction, to visualize better in terms of 2-D plot we have chosen the dimensions to be 2. PCA selects the 2 features with highest variance in each class. We plot this reduced data points in graph and observed the data-set. The resulting graph made it clear that while the data is easily separable between the two high level activities (Static and dynamic), the separability between classes in the same cluster was very less. We also observed that the data-set would be better classified a simpler classifier as the classes were in distinct clusters.

12.1.2 Dropping unimportant features

We ran the ExtraTreesClassifier to find out the importance of features and their ranking. We obtained a graph which is shown below. We decided to drop the un-important features and we progressive dropped 250 , 200 , 150, 100 , and 61 features to check the accuracy of all the classifiers. We found out that all the models performed optimum when we dropped the least important 61 features. This led to another interesting observation that the co-relation between features is minimal hence the higher accuracy at increasing the no of features.

12.2 Comparison of Results

Comparision	
Classifier Algorithm	**Accuracy**
Linear Discriminant Analysis	96.33%
Quadratic Discriminant Analysis	78.65%
Random Forest	92.77%
Perceptron Neural Network	94.60%
Support Vector Classifier [Polynomial kernel]	92.56%
Support Vector Classifier [linear kernel]	96.43%
Support Vector Classifier [RBF kernel]	95.04%
Random Forest	92.77%
K-Nearest Neighbor	87.75%
Bagging Classifier using SVM Linear kernel	95.96%
Gradient Boosting Classifier	93.48%

We initially started with SVM using the default RBF kernel and the accuracy was very less so we shifted our attention to Gaussian discriminant analysis and its implementations in sklearn, linear discriminant analysis and quadratic discriminant analysis. The structure of data and the previous experimentation led us to believe that our data was more suited to simple classifiers. The idea of GDA seemed simple and intuitive enough and as expected we got higher accuracy by Linear discriminant analysis. In the end we applied the same SVM classifier using a linear kernel which resulted in the highest accuracy for our model.

12.3 Limitations of current work

Though we we able to achieve reasonable accuracy with current dataset and different classifiers, the analysis and prediction models are heavily constrained by the limited size of the training set. A more robust dataset would enable application of complicated models like neural networks and prediction of more activities. Some more problems that we came across are

1. Though we didn't do the pre-processing of data,the processing of continuous data would increase the latency of the system. This rules out real-time prediction, though mobile hardware has improved on a exponential scale and this observation needs to be verified on current hardware.

2. Discrimination between activities is a very important aspect that is major bottleneck for scaling the experiment to include other activities. As the no of activities increase so does the possibility of an overlap between different activities.

Chapter 13

Contribution of Team members

1. Abhishek bagade 25% contribution
   Worked on different models and feature engineering

2. Rahul Anand 25% contribution
   Worked on different models and generating graphs.

3. Jitendra Bafna 25% contribution
   Compiling resources and reading the theoretical basis for the papers. Worked to study the data and pre-processing.

4. Sudhanshu Deshmukh 25% contribution
   Worked on the report and seminar organization and compilation of data. Worked on searching for the librarires to use.

Bibliography

[1] E. Kim, S. Helal, and D. Cook. Human activity recognition and pattern discovery. *IEEE Pervasive Computing*, 9(1):48–53, 2010. cited By 180.

[2] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J.L. Reyes-Ortiz. Energy efficient smartphone-based activity recognition using fixed-point arithmetic. *Journal of Universal Computer Science*, 19(9):1295–1314, 2013. cited By 20.

[3] J.L. Reyes-Ortiz, A. Ghio, D. Anguita, X. Parra, J. Cabestany, and A. Catal. Human activity and motion disorder recognition: Towards smarter interactive cognitive environments. pages 403–412, 2013. cited By 1.

[4] J. T. Sunny, S. M. George, and J. J. Kizhakkethottam. Applications and challenges of human activity recognition using sensors in a smart environment. *International Journal for Innovative Research in Science and Technology*, 2(4):50–57, 2015.

[5] .D. Lara and M.A. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, 15(3):1192–1209, 2013. cited By 225.

[6] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J.L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. pages 437–442, 2013. cited By 51.