Human Activity Recognition

Introduction

Understanding the physical state of a person's body from simply a smartphone's accelerometer and gyroscope can give us insight to understand what particular feature or motion is helpful to determine different states of a person's actions such as walking, walking up or down, standing, sitting and laying.

Using exploratory analysis and two predictive models, it has been shown that there are significant differences between the six physical states. The models were run on the Human Activity Recognition dataset were subjects were tasked to perform the six different actions and their motions were captured. This report describes the data in more detail and how classification was done along with its corresponding results. Classification has been carried out by varying the number of dimensions in the data to determine the importance of the dimensions in the data.

Data Collection and Preparation

The Human Activity Recognition data was provided through the UCI Machine Learning repository. In this study 30 subjects were recruited to perform six key tasks while wearing Samsung Galaxy smartphone on the waist. The tasks were walking, walking upstairs, walking downstairs, sitting, standing, and laying. The features collected from this study were generated from raw accelerometer and gyroscope signals. The embedded accelerometer and gyroscope were used to capture 3- axial linear acceleration and 3 axial angular velocity at a constant rate of 50Hz.

The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 seconds and 50% overlap (128 data points / window). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The acceleration signal was then separated into body and gravity acceleration signals. From each window, a vector of features was obtained by calculating variables from the time and frequency domain. Hence, for each time segment of 2.56 seconds, 561 features were extracted

Exploratory Analysis

To understand and visualize the data distribution better dimensionality reduction was using t-Distributed Stochastic Neighbor Embedding (t-SNE). t-Distributed Stochastic Neighbor Embedding is a non-linear dimensionality reduction algorithm used for exploring high-dimensional data. It maps multi-dimensional data to two or more dimensions suitable for human observation. Since the Human Activity Recognition Dataset contains 561 dimensions this technique is useful to see if there are clear and distinct regions in the data by reducing it to two dimensions. Below you can see a scatter plot after applying t-SNE on the Human Activity Recognition dataset.

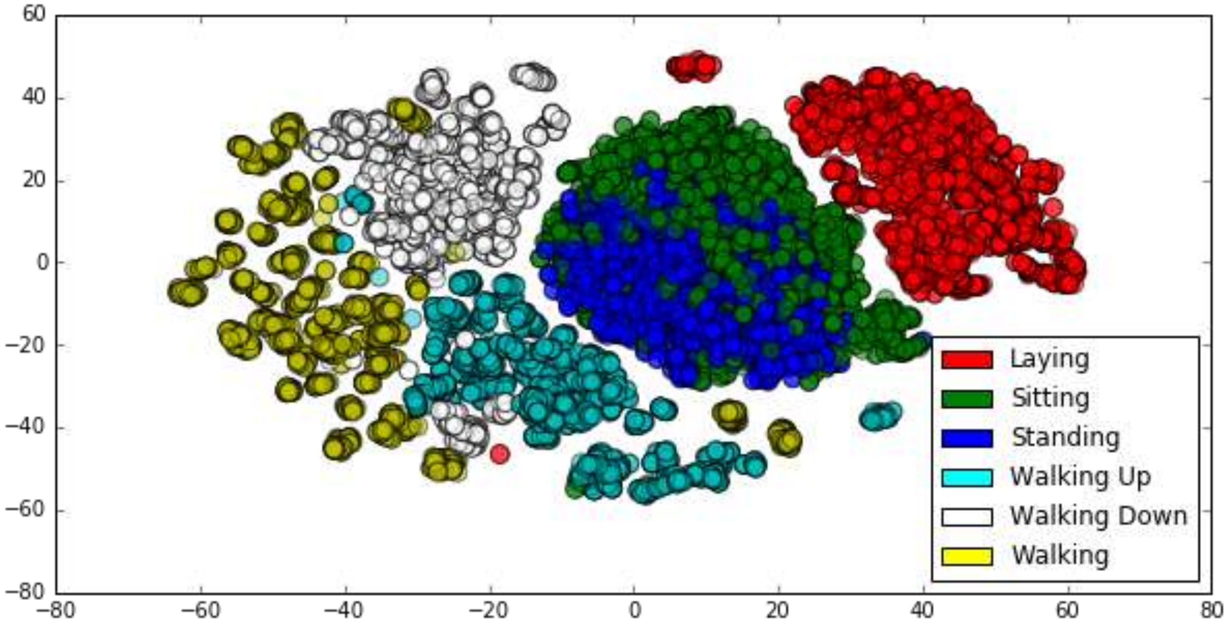

Fig 1: Scatter plot of HAR data after dimensionality reduction with t-SNE

From Fig 1, it the six regions (six classes) can be found to be quite distinct, meaning it should be easy to classify them. The only concern is among the blue and green regions corresponding to Standing and Sitting respectively, since there is a lot of overlap. It may be quite difficult to classify those regions.

It was also important to look at the class distribution of the different activities in the dataset, since it would play a vital role measuring the accuracy of the classification models. Table 1 below shows the distribution of the six different classes in the dataset.

Activity	Distribution
Walk Down	13.65 %
Walk Up	14.99 %
Sitting	17.25%
Standing	18.51%
Walking	16.72 %
Laying	18.88%

Table: 1

From the table above, it can be seen that the class distribution is quite balanced. Hence, while running the classification model accuracy can be used as a proper metric ass there is no significant class imbalance in the dataset.

The correlation between the features were also analyzed. It was used to find which dimensions were highly correlating with each other to understand which features may appear together to anticipate potential problems while classifying. Below listed are the 61 features that have a correlation lower than 0.5 with each other.

```
'tBodyAcc-mean()-X', 'tBodyAcc-mean()-Y', 'tBodyAcc-mean()-Z', 'tBodyAcc-
arCoeff()-X,4', 'tBodyAcc-arCoeff()-Y,4', 'tBodyAcc-arCoeff()-Z,4',
'tBodyAcc-correlation()-X,Y', 'tBodyAcc-correlation()-X,Z', 'tGravityAcc-
entropy()-Y', 'tGravityAcc-entropy()-Z', 'tGravityAcc-correlation()-X,Y',
'tGravityAcc-correlation()-X,Z', 'tBodyAccJerk-mean()-X', 'tBodyAccJerk-
mean()-Y', 'tBodyAccJerk-mean()-Z', 'tBodyAccJerk-arCoeff()-Y,4',
'tBodyAccJerk-arCoeff()-Z,4', 'tBodyAccJerk-correlation()-X,Y',
'tBodyAccJerk-correlation()-X,Z', 'tBodyAccJerk-correlation()-Y,Z',
'tBodyGyro-mean()-Z', 'tBodyGyro-arCoeff()-X,4', 'tBodyGyro-arCoeff()-
Y,4', 'tBodyGyro-arCoeff()-Z,4', 'tBodyGyro-correlation()-X,Y',
'tBodyGyro-correlation()-X,Z', 'tBodyGyro-correlation()-Y,Z',
'tBodyGyroJerk-mean()-X', 'tBodyGyroJerk-mean()-Y', 'tBodyGyroJerk-mean()-
Z', 'tBodyGyroJerk-arCoeff()-X,4', 'tBodyGyroJerk-arCoeff()-Y,4',
'tBodyGyroJerk-arCoeff()-Z,4', 'tBodyGyroJerk-correlation()-X,Y',
'tBodyGyroJerk-correlation()-X,Z', 'tBodyGyroJerk-correlation()-Y,Z',
'tBodyAccJerkMag-arCoeff()3', 'tBodyAccJerkMag-arCoeff()4',
'tBodyGyroJerkMag-arCoeff()3', 'tBodyGyroJerkMag-arCoeff()4', 'fBodyAcc-
maxInds-Y', 'fBodyAcc-maxInds-Z', 'fBodyAcc-kurtosis()-Z', 'fBodyAccJerk-
maxInds-Y', 'fBodyGyro-maxInds-X', 'fBodyGyro-maxInds-Z', 'fBodyGyro-
skewness()-X', 'fBodyGyro-kurtosis()-X', 'fBodyGyro-skewness()-Y',
'fBodyGyro-kurtosis()-Y', 'fBodyGyro-skewness()-Z', 'fBodyGyro-kurtosis()-
Z', 'fBodyBodyAccJerkMag-maxInds', 'fBodyBodyGyroMag-maxInds',
'fBodyBodyGyroMag-skewness()', 'fBodyBodyGyroMag-kurtosis()',
'fBodyBodyGyroJerkMag-maxInds', 'angle(tBodyAccMean,gravity)',
'angle(tBodyAccJerkMean),gravityMean)',
'angle(tBodyGyroJerkMean,gravityMean)'
```

The dataset contains tasks done by 30 individuals, however, they repeated their tasks, so the dataset contains datapoints which come from the same individuals. This was important to keep in mind while developing the prediction model. During testing and training, if the data was split simply by each row and not by the individuals, the algorithm would have trained with bias. The movement of the same person is likely to be similar while performing a task. Hence, if the algorithm was tested on a datapoint from the same person whose datapoint it was trained on there would be bias in the accuracy of the predictions.

Methods

Objective 1

Logistic Regression and Support Vector Classifier classification models were used to predict what kind of activity was being done by the subjects. In the beginning of the model, the data is divided in test and train sets by making division across the subjects and not the individual datapoints. The test set is made of datapoints from randomly selected 4 subjects and the training set is made from the rest 26 subjects of the Human Activity Recognition dataset. Hence, the train set contained data from the same subjects and likewise with the testing set. During training, in the similar way a development test set is created from a subset of the subjects in the training set. For the development set datapoints from one-fourth of the subjects in the training set are taken.

The development test set is used to randomly select the best hyper-parameter for the algorithm. From a set of hyper-parameter values the model is trained and tested on the development set. The model with the hyper-parameter that results in the best accuracy is chosen as the best model. For Logistic regression in addition to hyperparameter tuning, a l1 regularization was used in its cost function. In the case of Support Vector Classifier, in addition to hyperparameter tuning, a radial basis function kernel was used while training the model.

Objective 2

Support Vector Classifier was used to vary the number of features from 100 to 560, with an interval of 5. The SelectKBest technique from the sklearn feature selection package was used to select the best features from the given number at every iteration. In this model the SelectKBest technique uses a chi-square statistic between a particular feature and the final classification. Hence, a small value most likely means that the feature is independent of the classification, whereas a high value is more likely to be important in the classification. Therefore, this model scores all the features in order of importance and uses the k (selected number of features to be chosen) highest scored features to train the model.

To increase the robustness of the model, while testing and training the dataset, stratified cross validation was applied with 10 folds for every iteration of feature selection. Hence, while reporting the accuracy values for every set of selected features, the average accuracy from all the folds were taken.

Analysis and Results

<u>Support Vector Classifier (SVC)</u>

Test Accuracy	0.97
Train Accuracy	1.00

Table 2: Accuracy of test and train on SVC

		Predicted Class					
		Standing	Sitting	Laying	Walking	Walking Downstairs	Walking Upstairs
Actual Class	Standing	214	16	0	0	0	0
	Sitting	11	194	2	0	0	0
	Laying	0	0	217	0	2	0
	Walking	0	0	0	267	0	0
	Walking Downstairs	0	0	0	0	184	0
	Walking Upstairs	0	0	0	2	1	195

Table 3: Confusion Matrix on test set from SVC

In this section we analyze the results from the Support Vector Classifier and Logistic Regression Classifier models. For Objective 1, both models have been compared using confusion matrix from the test set and the accuracy levels of the model on both the testing and training dataset

In Table 2 we find the accuracy of the SVC model using all features to be 97%, and for training set it was able to classify the data with a 100% accuracy. Hence, it can be confidently concluded that the model does not overfit as accuracy from both train and test are quite close to each other. To take a closer look at the results from prediction using SVC, Table 3 can give a clearer picture. Table 3 is a confusion matrix showing the true classification and misclassification for each class in the data. While most classes had a really low misclassification, 'Standing' and 'Sitting' were the two classes that had the most misclassification among each other. It makes sense, if you refer to Fig 1 which was a t-SNE plot of the Human Activity Recognition data. The Sitting and Standing (green and blue points) were the two classes that did not have a clear distinction among themselves, hence, it is quite difficult for the prediction model to classify them properly.

Logistic Regression

Test Accuracy	0.97
Train Accuracy	0.99

Table 4: Accuracy on test and train data for Logistic Regression

		Predicted Class					
		Standing	Sitting	Laying	Walking	Walking Downstairs	Walking Upstairs
Actual Class	Standing	219	11	0	0	0	0
	Sitting	16	188	1	0	0	2
	Laying	0	2	217	0	0	0
	Walking	0	0	0	267	0	0
	Walking Downstairs	0	0	0	0	184	0
	Walking Upstairs	0	0	0	7	0	191

Table 5: Confusion Matrix of test set from Logistic Regression

In the case of the Logistic Regression model the accuracy on the test set data was 97%, which is the same as what was found on SVC. Even though logistic regression tries to find the best hyperplane that best separates the classes, and SVC uses a similar method to find a hyperplane to make distinctions among the classes. Since from Fig 1, it can be seen that the data was quite separable, the radial basis function used in SVC, did not result in making a non-linear hyperplane. Hence, it is fair to assume that both the SVC and Logisitic Regression model used a similar linear hyperplane to classify the data, resulting in similar prediction accuracies.

From Table 5, it is clear to see that the confusion matrix from Logistic Regression is similar to that in Table 3 for SVC. The relatively highest miss classification also exists among Standing and Sitting. It is also worthwhile to point out that Logistic Regression also miss classified Walking Upstairs to be Walking for 7 cases as well. Since, both activities are quite similar the data from the features may be quite similar for some cases.

For the second objective, in this section the accuracies resulted from varying the number of features using the SVC model has been plotted in Fig 2. Notice, that the accuracy across features is roughly between 0.9 and 0.95, which is lower than the accuracy found in the previous section in both models (logistic regression and SVC). Such a result is expected as the accuracies are average accuracies from applying ten-fold cross-validation. Hence, it results in more robust results.

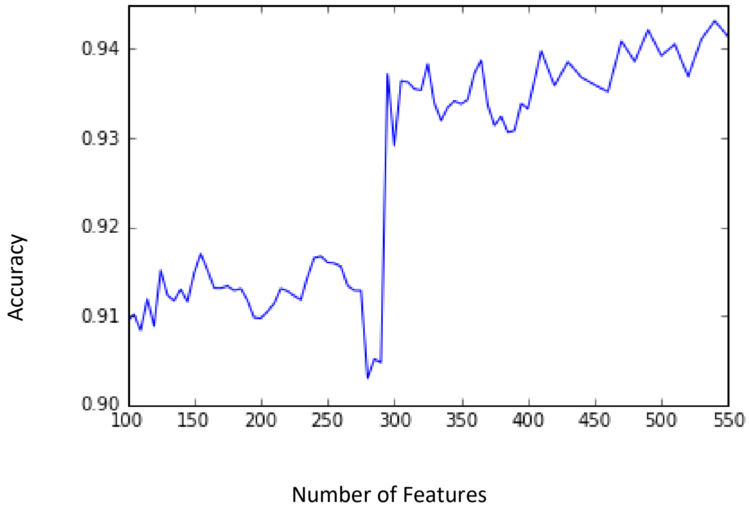

Fig 2: Accuracy values as features varies

Fig 2 shows that there is a general trend of increase in accuracy values as the number of features increase. However, the rise in accuracy is quite low. Even though the SelectKBest technique selects the best features for classification for the given number of features, this result tells us that other features may still have some type of information that can improve classification a little more. It is also quite interesting to notice the sudden spike in accuracy value between 270 to 290 features being used.

Conclusions

The Analysis suggests that it is not difficult to classify Human Activity in this dataset. Since, both models produced similar results and bias was removed as much as possible by keeping sperate test sets along with keeping subjects in both test and train set separate. Most of the 561 features contains quite important information to classify Human Activity in this dataset as increasing features showed increase in accuracy levels.