ECS 330 MINI LAB PROJECT

Smartphone Based Recognition of Human Activities

Project ID –17

Faculty Advisor- Kundan Khandway

Name: Tanishq Kumar Baswal

Roll No: 18291

❖ ABSTRACT

Human-centered computing is an emerging research field that aims to understand human behaviour and integrate users and their social context with computer systems. One of the most recent, challenging and appealing applications in this framework consists in sensing human body motion using smartphones to gather context information about people actions. In this context, we describe in this work an Activity Recognition database, built from the recordings of 30 subjects doing Activities of Daily Living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors, which is released to public domain on a well-known on-line repo- itory. Results, obtained on the dataset by exploiting a multiclass Support Vector Machine (SVM), are also acknowledged.

❖ INTRODUCTION

Human Activity Recognition (HAR) aims to identify the actions carried out by a person given a set of observations of him/herself and the surrounding environment. Recognition can be accomplished by exploiting the information retrieved from various sources such as environmental or body-worn sensors. Some approaches have adapted dedicated motion sensors in different body parts such as the waist, wrist, chest and thighs achieving good classification performance. These sensors are usually uncomfortable for the common user and do not provide a long-term solution for activity monitoring (e.g., sensor repositioning after dressing).

Smartphones are bringing up new research opportunities for human-centered applications where the user is a rich source of context information and the phone is the first-hand sensing tool. Latest devices come with embedded built-in sensors such as microphones, dual cameras, accelerometers, gyroscopes, etc. The use of smartphones with inertial sensors is an alternative solution for HAR. These mass-marketed devices provide a flexible, affordable and self-contained solution to automatically and unobtrusively monitor Activities of Daily Living (ADL) while also providing telephony services. Consequently, in the last few years, some works aiming to understand human behaviour using smartphones have been proposed:

❖ DATASET ANALYSIS

http://archive.ics.uci.edu/ml/datasets/SmartphoneBased+Recognition+of+Human+Activities+and+Postur-al+Transitions

A set of experiments were carried out to obtain the HAR dataset. A group of 30 volunteers with ages ranging from 19 to 48 years were selected for this task. Each person was instructed to follow a protocol of activities while wearing a waist-mounted Sam- sung Galaxy S II smartphone. The six selected ADL were *standing, sitting, laying down, walking, walking downstairs and upstairs*. Each subject performed the protocol twice: on the first trial the smartphone was fixed on the left side of the belt and on the second it was placed by the user himself as preferred. There is also a separation of 5 seconds between each task where individuals are told to rest, this facilitated repeatability (every activity is at least tried twice) and ground trough generation through the visual interface. The tasks were performed in laboratory conditions but volunteers were asked to perform freely the sequence of activities for a more naturalistic dataset.

The experiment also included postural transitions that occurred between the static postures: stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie, lie-to-stand.

The obtained dataset was randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data.
The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/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 gravitational force is assumed to have only low frequency components, therefore a filter with 0.3 Hz cut-off frequency was used. From each window, a vector of 561 features was obtained by calculating variables from the time and frequency domain.

The dataset is divided in two parts-

1. Inertial Sensor Data:

- Raw triaxial signals from the accelerometer and gyroscope of all the trials with participants.

- The labels of all the performed activities.

2. Records of activity windows:

- A 561-feature vector with time and frequency domain variables.

- Its associated activity label.

- An identifier of the subject who carried out the experiment.

- Features are normalized and bounded within [-1,1].

- The units used for the accelerations (total and body) are 'g'(acceleration due to gravity-> 9.80665 m/sec2).

- The gyroscope units are rad/sec.

Which Features Are There?

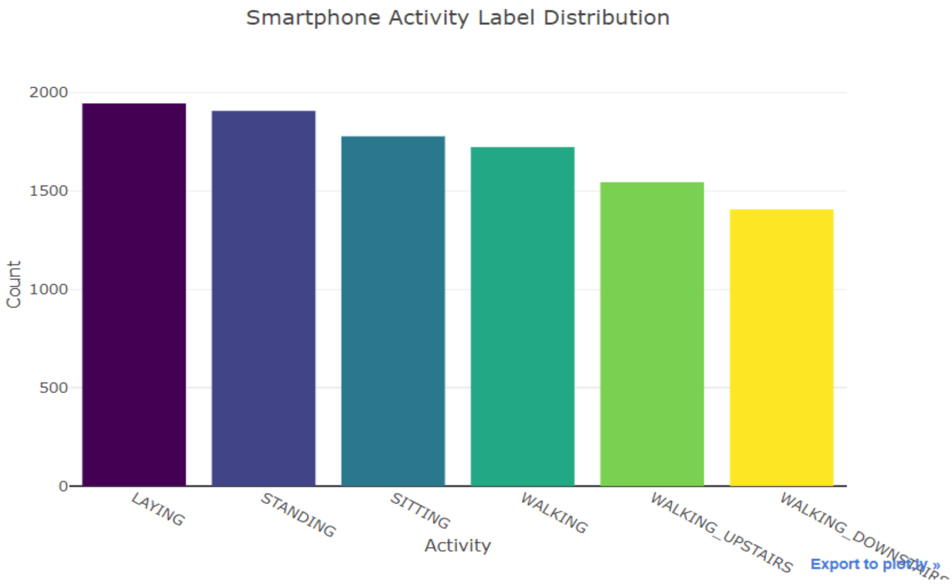
The features seem to have a main name and some information on how they have been computed attached. Grouping the main names will reduce the dimensions for the first impression.

```
# Group and count main names of columns
pd.DataFrame.from_dict(Counter([col.split('-')[0].split('(')[0] for col in both_df.column
s]), orient='index').rename(columns={0:'count'}).sort_values('count', ascending=False)
```

	count
fBodyAcc	79
fBodyGyro	79
fBodyAccJerk	79
tGravityAcc	40
tBodyAcc	40
tBodyGyroJerk	40
tBodyGyro	40
tBodyAccJerk	40
tBodyAccMag	13
tGravityAccMag	13
tBodyAccJerkMag	13
tBodyGyroMag	13
tBodyGyroJerkMag	13
fBodyAccMag	13
fBodyBodyAccJerkMag	13
fBodyBodyGyroMag	13
fBodyBodyGyroJerkMag	13
angle	7
subject	1
Data	1

How Labels Are Distributed?

```
# Plotting data
label_counts = label.value_counts()

# Get colors
n = label_counts.shape[0]
colormap = get_cmap('viridis')
colors = [rgb2hex(colormap(col)) for col in np.arange(0, 1.01, 1/(n-1))]

# Create plot
data = go.Bar(x = label_counts.index,
              y = label_counts,
              marker = dict(color = colors))

layout = go.Layout(title = 'Smartphone Activity Label Distribution',
                   xaxis = dict(title = 'Activity'),
                   yaxis = dict(title = 'Count'))

fig = go.Figure(data=[data], layout=layout)
iplot(fig)
```



Although there are fluctuations in the label counts, the labels are quite equally distributed.

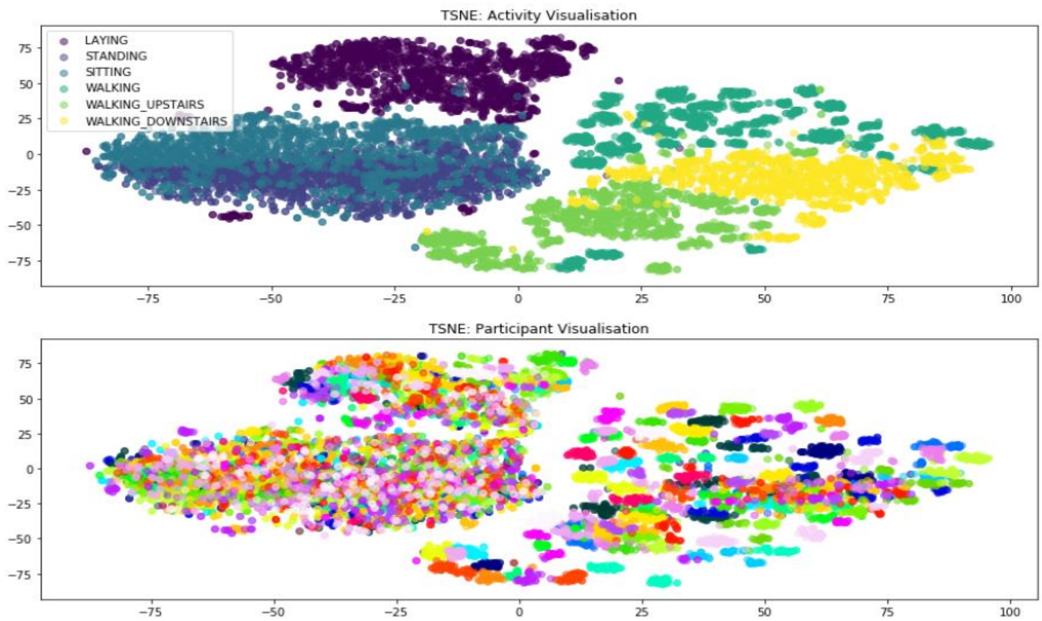
Assuming the participants had to walk the same number of stairs upwards as well as downwards and knowing the smartphones had a constant sampling rate, there should be the same amount of datapoints for walking upstairs and downstairs. Disregarding the possibility of flawed data, the participants seem to **walk roughly 10% faster downwards.**

Are The Activities Separable?

The dataset is geared towards classifying the activity of the participant.

```
# Create datasets
tsne_data = both_df.copy()
data_data = tsne_data.pop('Data')
subject_data = tsne_data.pop('subject')

# Scale data
scl = StandardScaler()
tsne_data = scl.fit_transform(tsne_data)

# Reduce dimensions (speed up)
pca = PCA(n_components=0.9, random_state=3)
tsne_data = pca.fit_transform(tsne_data)

# Transform data
tsne = TSNE(random_state=3)
tsne_transformed = tsne.fit_transform(tsne_data)


# Create subplots
fig, axarr = plt.subplots(2, 1, figsize=(15,10))

### Plot Activities
# Get colors
n = label.unique().shape[0]
colormap = get_cmap('viridis')
colors = [rgb2hex(colormap(col)) for col in np.arange(0, 1.01, 1/(n-1))]
```

```
# Plot each activity
for i, group in enumerate(label_counts.index):
    # Mask to separate sets
    mask = (label==group).values
    axarr[0].scatter(x=tsne_transformed[mask][:,0], y=tsne_transformed[mask][:,1], c=colors[i],
alpha=0.5, label=group)
axarr[0].set_title('TSNE: Activity Visualisation')
axarr[0].legend()


### Plot Subjects
# Get colors
n = subject_data.unique().shape[0]
colormap = get_cmap('gist_ncar')
colors = [rgb2hex(colormap(col)) for col in np.arange(0, 1.01, 1/(n-1))]

# Plot each participant
for i, group in enumerate(subject_data.unique()):
    # Mask to separate sets
    mask = (subject_data==group).values
    axarr[1].scatter(x=tsne_transformed[mask][:,0], y=tsne_transformed[mask][:,1], c=colors[i],
alpha=0.5, label=group)

axarr[1].set_title('TSNE: Participant Visualisation')
plt.show()
```



In plot 1 you can clearly see the **activities are mostly separable.**

Plot 2 reveals **personal information** of the participants. Everybody has for example an **unique/sparable walking style** (on the upper right). Therefore, the smartphone should be able to **detect what you are doing and also who is using the smartphone** (if you are moving around with it).

❖ EXPERIMENTAL RESULTS

For this purpose, we exploit well-known and state-of-the-art Support Vector Machine (SVM) binary classifiers, which are generalized to the multiclass case through a One-Vs-All (OVA) approach: the SVM hyperparameters are selected through Cross Validation procedure and kernels are selected on best score on training dataset.

Importing libraries and loading dataset

```
# Commented out IPython magic to ensure Python compatibility.
import pandas as pd
import numpy as np
import numpy as np
import pylab as pl
import pandas as pd
import matplotlib.pyplot as plt
# %matplotlib inline

from sklearn.utils import shuffle
from sklearn.svm import SVC
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix,classification_report
from sklearn.model_selection import cross_val_score, GridSearchCV

#LOADING DATASET
train_df = pd.read_csv('train.csv')
test_df = pd.read_csv('test.csv')
print('Shape Train:\t{}'.format(train_df.shape))
print('Shape Test:\t{}\n'.format(test_df.shape))
train_df.head()

```

Information About Labels

Keeping count of each Label

```
# INFORMATION ABOUT LABELS
train_outcome = pd.crosstab(index=train_df["Activity"],  # Make a crosstab
                            columns="count")      # Name the count column

```

col_0	count
Activity	
LAYING	1407
SITTING	1286
STANDING	1374
WALKING	1226
WALKING_DOWNSTAIRS	986
WALKING_UPSTAIRS	1073

Normalize the Predictor (Feature Set) for SVM training

```
# MODELLING
X_train = pd.DataFrame(train_df.drop(['Activity','subject'],axis=1))
Y_train_label = train_df.Activity.values.astype(object)
X_test = pd.DataFrame(test_df.drop(['Activity','subject'],axis=1))
Y_test_label = test_df.Activity.values.astype(object)

#ENCODING
encoder = preprocessing.LabelEncoder()
encoder.fit(Y_train_label)
Y_train = encoder.transform(Y_train_label)
encoder.fit(Y_test_label)
Y_test = encoder.transform(Y_test_label)

# SCALING
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Hyperparameter tuning using grid search and cross validation

```
# CHOOSING THE BEST KERNEL FOR THE MODEL
params_grid = [{'kernel': ['rbf'], 'gamma': [1e-3, 1e-4],
                     'C': [1, 10, 100, 1000]},
                    {'kernel': ['linear'], 'C': [1, 10, 100, 1000]}]
```

Training SVM model using radial kernel

```
svm_model = GridSearchCV(SVC(), params_grid, cv=5)
svm_model.fit(X_train_scaled, Y_train)

print('Best score for training data:', svm_model.best_score_,"\n")
print('Best C:',svm_model.best_estimator_.C,"\n")
print('Best Kernel:',svm_model.best_estimator_.kernel,"\n")
print('Best Gamma:',svm_model.best_estimator_.gamma,"\n")
```

```
Best score for training data: 0.938255432696532

Best C: 1000

Best Kernel: rbf

Best Gamma: 0.0001
```

Confusion Matrix and Accuracy Score

```
# CONFUSION MATRIX
final_model = svm_model.best_estimator_
Y_pred = final_model.predict(X_test_scaled)
Y_pred_label = list(encoder.inverse_transform(Y_pred))

print(confusion_matrix(Y_test_label,Y_pred_label))
print("\n")
print(classification_report(Y_test_label,Y_pred_label))

print("Training set score for SVM: %f" % final_model.score(X_train_scaled , Y_train))
print("Testing  set score for SVM: %f" % final_model.score(X_test_scaled  , Y_test ))

svm_model.score
```

```
[[537   0   0   0   0   0]
 [  0 436  54   0   0   1]
 [  0  15 517   0   0   0]
 [  0   0   0 493   3   0]
 [  0   0   0   5 398  17]
 [  0   0   0  16   2 453]]

                    precision    recall  f1-score   support

            LAYING       1.00      1.00      1.00       537
           SITTING       0.97      0.89      0.93       491
          STANDING       0.91      0.97      0.94       532
           WALKING       0.96      0.99      0.98       496
WALKING_DOWNSTAIRS       0.99      0.95      0.97       420
  WALKING_UPSTAIRS       0.96      0.96      0.96       471

          accuracy                           0.96      2947
         macro avg       0.96      0.96      0.96      2947
      weighted avg       0.96      0.96      0.96      2947

Training set score for SVM: 0.996872
Testing  set score for SVM: 0.961656
```

❖ CONCLUSION

Using SVM machine learning algorithm we are able to identify different activities stated in the dataset with accuracy of about 96.16 %.

❖ REFERENCES

1. Human Activity Recognition on Smartphones With Awareness of Basic Activities and Postural Transitions.(Jorge-Luis Reyes-Ortiz, Luca Oneto, Alessandro Ghio, Albert SamÃ¡, Davide Anguita and Xavier Parra).

Artificial Neural Networks and Machine Learning â€“ ICANN 2014. Lecture Notes in Computer Science. Springer. 2014.

2. A Public Domain Dataset for Human Activity Recognition Using Smartphones.(Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz).

21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.

3. Energy Efficient Smartphone-Based Activity Recognition using Fixed-Point Arithmetic.(Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge L. Reyes-Ortiz).

Journal of Universal Computer Science. Special Issue in Ambient Assisted Living: Home Care. Volume 19, Issue 9. May 2013

4. Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine.(Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz).

4th International Workshop of Ambient Assited Living, IWAAL 2012, Vitoria-Gasteiz, Spain, December 3-5, 2012.

Proceedings. Lecture Notes in Computer Science 2012, pp 216-223.

5.Human Activity and Motion Disorder Recognition: Towards Smarter Interactive Cognitive Environments. (Jorge Luis Reyes-Ortiz, Alessandro Ghio, Xavier Parra-Llanas, Davide Anguita, Joan Cabestany, Andreu CatalÃ).

21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.