Human Activity Recognition using Smart Phones

Maciej Najdecki, s12563
Krystian Rygiel, s12809
Krzysztof Nowicki, s12824

January 17, 2017

Contents

1 Task description

This report covers the classification of experiment results which was performed on group of 30 volunteers. Each person performed six activities (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz were captured. The experiments have been video-recorded to label the data manually.
The main purpose of this project is to perform **feature selection** and **classification** of the feature vectors. The experimental process of developing program and performing experiment has been widely described in further sections of this article.

2 Initial dataset

The dataset has been downloaded from the **UCI** repository from link assigned to this particular problem, the **Human Activity Recognition using Smart Phones**. The brief description of the data may be found there. Due to task requirements, we were supposed to train classifier on the entire dataset. The datasource was already splitted as two distinct non overlapping sets- training and testing one. Task requires to split entire set randomly, so first of all we have merged sets to the one containing all training and testing instances.

2.1 Number of records

After successful merging of training and testing instances to single datasource it contains **10299** instances in total. Due to dataset characteristics they reside as two separate files, one with feature vectors (*dataset/train_features.txt*) and second one with decision class indexes (*dataset/train_labels.txt*).

2.2 Number of attributes

Each instances is described by feature vector with length equal to **561**. The description of each feature meaning can be found on UCI webpage. Names and indexes of all features may be found in file (*dataset/features.txt*).

2.3 Decision classes distribution

The decision class of instance is one from the predefined set {WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING}.
There are **6** decision classes assigned to the instances. The labels are stored in 4th data file called (*dataset/activity_labels.txt*)

2.4 Preprocessing

Task is highly related with attributes filtering. We decided to use **WEKA** machine learning library to preprocess (filter out) redundant features. The library accepts only files with .arff format, so we wrote a converter in **Python 3.4** to create such file. The script (*scripts/arff_maker.py*) is responsible to merge four mentioned data files in to the single .arff one which could be further processed by WEKA library.
To perform the file concatenation following command must be run in shell terminal (assuming that current pwd is script folder). It is vital to redirect the output of the script to file with .arff extension, like:

```
./arff_maker.py \
   ../data/features.txt \
   ../data/activity_labels.txt \
   ../data/train_features.txt \
   ../data/train_labels.txt > ../datasets/dataset.arff
```

Above command will join files together and make the valid .arff datasource file which may be used as input to weka.jar to filter attributes.

3 Solving method

There were to possible approaches to solve the chosen task. The **K-nearest neighbours** alogrithm and **multilayer neural network** implementation. We have decided not to use ready to go classifiers from WEKA package but we developed our implementation of multilayer network which is learned by applying backpropagation algorithm. Due to fact that all of us are familiar with **Java** programming language we have decided to implement the solving algorithm in this environment. We developed a program solution that is fully configurable via command line arguments. All of **neural network** parameters

3.1 Dataset filtering

3.2 Feature selection

4 Classification algorithm

4.1 Implementation

4.2 Evaluation

4.3 Learning statistics

5 Experiment

5.1 Neural net architecture

5.2 Learning parameters

5.3 1 experiment - 8 features

5.4 2 experiment - 20 features

5.5 3 experiment - 50 features

5.6 4 experiment - 561 features

5.7 Conclusion