Human Activity Recognition based on Mobile Sensors

Abhishek Chandak, Mandar Munagekar, Nihal Mehta
Dept. Of Computer Science, California State University, Sacramento
abhishekchandak@csus.edu , mmunagekar@csus.edu, nihalnmehta@csus.edu

ABSTRACT

Smartphones being an integral part in our lives are not just restricted for calling and entertainment purposes but, their use case lie way beyond some of these basic applications. Since, the advent of Smartphones, a revolution has been created in the mobile communication industry. Enhancing user experience is the primary purpose of mobile communication industries and for this purpose they load the mobile phones with several sensors. Two of the such sensors are Accelerometer and Gyroscope. Accelerometer measures acceleration while Gyroscope measures angular velocity. Here, we will try to use the data provided by accelerometer and gyroscope of Smartphone to classify the activity which a Smartphone user is performing by using a dataset and classify the data as walking, walking-upstairs, walking-downstairs, sitting-down, standing-up and laying-down which formulates for a multi-class classification problem. [1]

Keywords: Smart Phones, Accelerometer, Gyroscope, Dataset, Classification.

1. Introduction

The experiments have been carried out with people performing different activities such as walking, walking-upstairs, walking-downstairs, sitting-down, standing-up and laying-down with the help of the embedded smartphone sensors. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data. These sensor signals are pre-processed by applying noise filters and then sampled in fixed-width windows.

2. Problem Statement

By using either human engineered 561 feature data or raw features of 128 reading, our goal is to predict one of the six activities that a Smartphone user is performing at 2.56 Seconds time window.

3. Description

No low latency requirement and errors which cost less are the primary objective and constraint of this problem statement which is addressed in the form of processing. Finally, after all this processing we get frequency domain signals from some of the available signals which are represented as features with names like tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag and tBodyGyroJerkMag. Based on this, further estimates of some set of variables are calculated and further, we obtain some other vectors by taking the average of signals in a single window sample. Some of these variables are mainly calculated based on mean. Accelerometer readings are divided into gravity acceleration and body acceleration readings, which has x, y and z components each. Gyroscope readings are the measure of angular velocities which has x, y and z components. Once we get a feature vector of 561 features, these features are given in the dataset. Each window of readings is a data-point of 561 features.

3.1 Dataset

The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. 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, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data.
Datasetlink:https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

4. Working

The Domain experts from the field of Signal Processing collects the data from Accelerometer and Gyroscope of Smartphone. These domain experts break up the data in the time window of 2.56 seconds with 50% overlapping. This data can be termed as raw data which is called 128 reading.

Fig.4.1: Workflow of Problem Statement

All the Accelerometer and Gyroscope readings are tri-axial, means that they measure acceleration and angular-velocity respectively in all the three axes namely X-axis, Y-axis and Z-axis. So, we have in total six time-series data. [3]
Given these six time-series data, we want to predict six activities. This is a multi-class classification problem wherein accuracy is one of the important metrics to be used and to further evaluate we will also use confusion-matrix to check that in which two activities our model is confused and predicting incorrect activity. For example, between standing-up and sitting-down.

5. Models Used and their Analysis

Tesseract-OCR is an open source tool used to recognize text in images. The process of extracting text from images is called as Optical Character Recognition (OCR). OCR is used to convert the text characters in digital images.[2] The basic process involves parsing the text and translating the characters into code that can be used for data processing and other purpose. OCR is also called as text recognition.[4] Optical scanner is used to read text and software handles the rest of the processing. To implement advanced or intelligent character recognition like identifying languages or styles of handwriting can also be done using OCR processing.

Developed by HP, tesseract OCR has many versions, and each had advancement and improvement from the others in many ways. Version 3 supports many image formats and gradually added many scripts (languages). It is based on traditional computer-vision algorithms. Computer Vision is the process of using devices to understand and analyze imagery. But recent advances in Machine Learning, and computing capabilities and high-quality input devices have major improvements. Computer vision is a classical application used for text recognition.

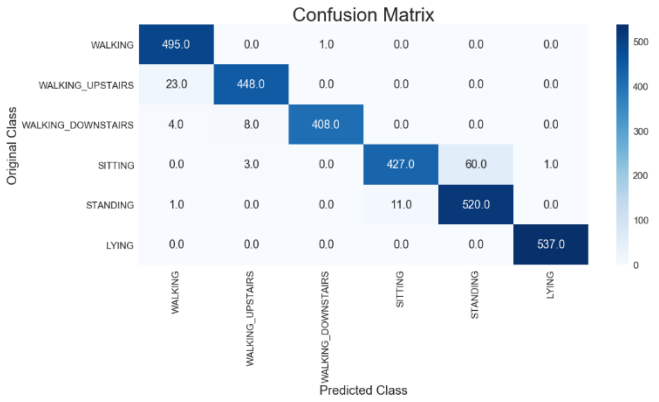
Empirical Evaluation of Models Implemented:

1. Logistic Regression:

   Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression).

   Accuracy Achieved: 96.2%

   Confusion Matrix:

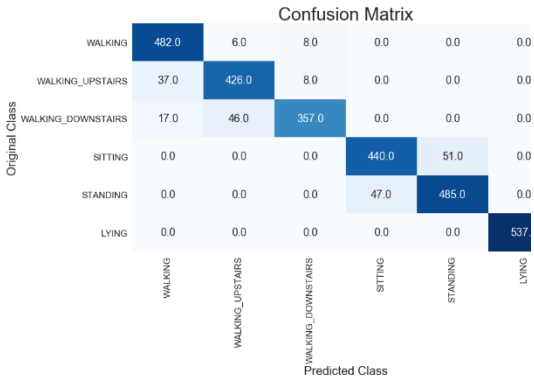


2. Random Forest:

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 or mean prediction of the individual trees.

Accuracy Achieved: 92.53%

Confusion Matrix:

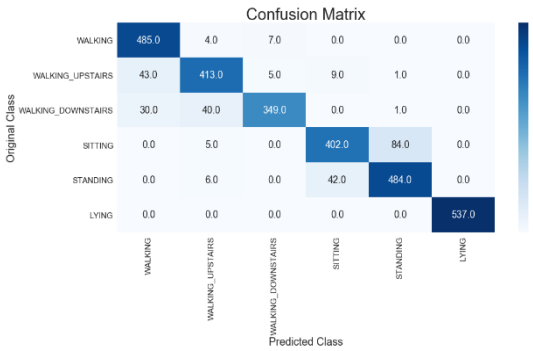


3. Gradient Boosted DT:

Gradient boosting. Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

Accuracy Achieved: 90.6%

Confusion Matrix:

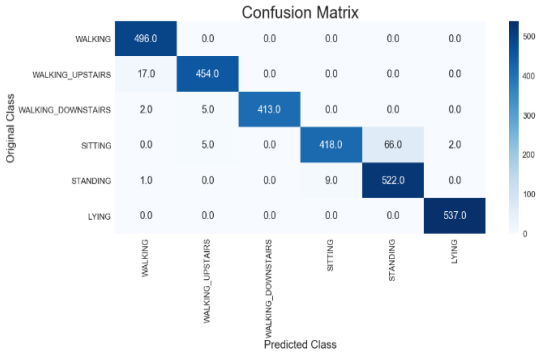


4. Linear SVM:

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

Accuracy Achieved: 96.37%

Confusion Matrix:

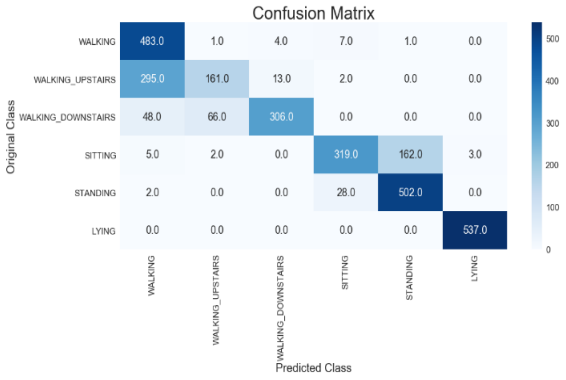


5. LSTM

Long Short-Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997) and were refined and popularized by many people in following work. They work tremendously well on a large variety of problems, and are now widely used.

Accuracy Achieved: 78.32%

Confusion matrix:



Page Layout Processing:

Tesseract inputs are binary images with optional polygonal text regions defined. The first step is, it is a connected component analysis and outlines are stored. The advantage of this is by inspection of the nesting of outlines, and number of sub outlines, it is simple to detect black-on-white text. Outlines are gathered together into blobs. These are organized into lines of text and regions are analyzed for fixed pitch or proportional text.

• Tesseract trials the text lines to determine whether they are fixed pitch. When it finds a fixed pitch text, tesseract chops the words into characters using the pitch, and disables the chopper. These text lines are

broken into words according to the kind of character spacing. The texts are broken into words using definite spaces and fuzzy spaces.[6]

• Non-fixed-pitch or proportional text spacing is a highly significant task. There are certain space or gap problems that have to be handled. Tesseract acts as a solution for most of these problems by measuring gaps in a limited vertical range between the baseline and mean line. Spaces that are close to the threshold at this stage are made uncertain, so that a final decision can be made after word recognition. Some difficult word spacing. The next step is recognition which is of two-pass process. The first is an attempt of recognizing each word. Each word is passed to a trained data. Then they accurately recognize text lower down the page since they may have learned. A second pass is done, run over the page, so the words not recognized can be processed again.

Line and Word Recognition:

The line finding algorithm is a design that a skewed page can be recognized without de-skewing, saving loss of image quality. It is the process of filtering blob and line construction. As the page layout analysis has already classified the region of text in a roughly uniform text size, a simple percentile height filter eliminates drop-caps and vertically touching characters. The median height provides the text size in the region, so it is safe to filter out blobs that are smaller than some fraction of the median height most likely punctuation, diacritical marks and noise. The filtered blobs fit a model of non-overlapping, parallel, but sloping lines. The blobs sorted and processed by x-coordinate makes it easy to assign blobs to a unique text line, while tracking the slope across the page, with greatly reduced danger of assigning to an incorrect text line. After the filtered blobs have been assigned to lines, a least median of squares fit is used to calculate the baselines, and the filtered-out blobs are fitted into the appropriate lines. The last step of the line creation process includes the process of merging blobs that overlap by at least half horizontally, putting diacritical marks together with the correct base, correctly associating parts of some broken characters. Next is a word recognition. The process of recognition for any character recognition engine is to identify how a word should be segmented into characters. The preliminary segmentation output from line finding is classified first. The rest of the word recognition applies to the to a non-fixed pitch text. For to get accurate results following can be also added to the processing like Chopping characters that are joined or continuous and associating broken characters, which has to be done to the characters that are chopped earlier.

Trained Data Classifier and Linguistic Analysis:

A classifier is used to recognize damaged characters easily, the classifier is not trained on damaged characters. The classifier is trained on a mere 20 samples of 94 characters from 8 fonts in a single size, and with attributes like normal, bold, italic, bold italic and others, total of 60160 training samples. Tesseract contains relatively little linguistic analysis. Whenever the word recognition module considers a new breakdown, the linguistic module chooses the best available word string in each of the following categories: frequently occurred word, dictionary word, numeric word, upper case word, lower case word, classifier choice word, etc. The final decision for a given segment is simply the word with the lowest total distance rating, where each of the above categories is multiplied by a different constant. Words from different segments may have different numbers of characters in them.

Static and Adaptive Classifier:

Static Character Classifier is one of the features used in early tesseract versions. But these features are not robust to the problems found in real life images. The alternate solution was the idea that the features in the unknown need not be the same as the features in the training data. While working out, the segments of a polygonal approximation are used for features, but in recognition, features of a small, fixed length are extracted from the outline and matched many-to-one contrary to the clustered prototype features of the training data.[5] It has been suggested that OCR engines can benefit from the use of an adaptive classifier. Since the static classifier must be good at generalizing to any kind of font, its ability to discriminate between different characters or between characters and non-characters is weakened. A more font-sensitive adaptive classifier that is trained by the output of the static classifier is therefore used to obtain greater discrimination within each document, where the number of fonts is limited.

Accuracy level:

Engine	Total char errors	Word Recall Errors	Word Precision Errors	*Wall time*	CPU time*
Tess 3.04	13.9	30	31.2	3.0	2.8
Cube	15.1	29.5	30.7	3.4	3.1
Tess+ Cube	11.0	24.2	25.4	5.7	5.3
LSTM	7.6	20.9	20.8	1.5	2.5

Table 4.1: Accuracy Level

Note in the above table illustrates that LSTM is faster than Tess 3.04.

6. Exploratory Data Analysis

Feature extraction from domain knowledge is mainly divided into two main types static and dynamic. In the case where test subject is in rest the features are sitting, standing, lying and when the test subject is in motion the dynamic features are walking, walking downstairs and walking upstairs.

Throughout the evaluation and running of the model we lay out a number of observations in order to keep a check on what is being performed and whether or not they conform with what is expected from the model which serves as a vital indicator in reaching our end goal.

7. Conclusion and Future work

We applied classical Machine Learning models on these 561 sized domain experts engineered features. LSTM works well on time-series data and thus, we decided that we will apply LSTM of Recurrent Neural Networks on 128 sized raw readings that we obtained from accelerometer and gyroscope signals. Using LSTM, we obtained a good accuracy. The Deep learning model help us to build models even when we do not have domain expert engineered features.

To further experiment, different machine learning models can be implemented. LSTM model can be further improved by running it for more epochs and more evaluations while tuning hyper-parameters such as activation function, optimizer and changing number of neurons.

8. References

[1] K. Y. C. Zaw Zaw Htike, Simon Egerton, "Real-time Human Activity Recognition using External and Internal Spatial Features," 2010 Sixth International Conference on Intelligent Environments, pp. 25-28, Jul.2010.

[2] C. Wu, A. H. Khalili, and H. Aghajan, "Multiview activity recognition in smart homes with spatio-temporal features," Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras - ICDSC '10, p. 142, 2010.

[3] V. Könönen, J. Mäntyjärvi, H. Similä, J. Pärkkä, and M. Ermes, "Automatic feature selection for context recognition in mobile devices," Pervasive and Mobile Computing, vol. 6, no. 2, pp. 181-197, 2010.

[4] J. Yang, "Toward Physical Activity Diary: Motion Recognition Using Simple Acceleration Features with Mobile Phones," Data Processing, pp. 1-9, 2009.

[5] T. S. Saponas, J. Lester, J. Froehlich, J. Fogarty, and J. Landay, "iLearn on the iPhone: Real-Time Human Activity Classification on Commodity Mobile Phones."

[6] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge L. Reyes-Ortiz. Energy Efficient Smartphone-Based Activity Recognition using Fixed-Point Arithmetic. Journal of Universal Computer Science. Special Issue in Ambient Assisted Living: Home Care. Volume 19, Issue 9. May 2013.