Machine Learning Assignment 1
Final Report

Group 03: Eklavya Sharma, Anirudh Kumar Bansal

October 24, 2017

Abstract

We describe machine learning models which take readings from a chest-mounted accelerometer and predict the type of activity the wearer is engaged in. We obtained labeled accelerometer readings data from UCI machine learning repository. We have studied two approaches for predicting activities. The first approach is training a Recurrent Neural Network (RNN). The second approach consists of extracting features from a sliding window and using them to train a classifier. We trained Random forest, Linear SVM and Logistic Regression. We obtained accuracy of around 93% with all of them.

Contents

1 Dataset Description

The dataset comprises of data from 15 participants doing 7 activities from a wearable accelerometer mounted on the chest. The accelerometer data is uncalibrated and sampled at 52 Hz.

The data is stored in 15 CSV files with one file for each participant. Each file stores the accelerometer readings while doing 7 activities. Each file contains around 100000 to 160000 samples (rows). Each sample has these comma-separated values:

- `sequence_number`
- `x_acceleration`
- `y_acceleration`
- `z_acceleration`
- `label`

`label` is a number which represents the type of activity. These are the possible values it can take:

- 1 - Working at computer
- 2 - Standing up, Walking and Going up-down stairs
- 3 - Standing
- 4 - Walking
- 5 - Going up-down stairs
- 6 - Walking and talking with someone
- 7 - Talking while standing

The x, y and z acceleration values are given as real numbers.

2 Methodology

2.1 Recurrent Neural Networks

A Recurrent Neural Network (RNN) consists of a neural network where one of the outputs of a layer is fed back as an input to the same or a previous layer. This creates a feedback loop. This leads to creation of a neural network where the output is affected not only by the input at a given time but also by previous inputs. This model is therefore suitable for classification tasks involving time-series analysis.

An RNN is generally trained by using Backpropagation Through Time (BPTT). RNNs (actually big neural networks in general) require a lot of training data. Data for 10 people will give us around a million samples, so we might have enough data.

RNNs don't work well in practice when temporal dependence is very long-term. For this, we need a different version of RNN, called a Long-Short-Term Memory Neural Network (LSTM NN).

On analyzing data, it was found that frequency of walking and stairs was around 25 Hz. This is long enough for RNN to fail. Therefore, LSTMs [2] were required. However, implementing or using LSTMs is a cumbersome and difficult process, so we think it's beyond the scope of this assignment.

2.2 Extracting features from sliding windows

To guess a person's activity at a particular time, we don't need to know their complete history. A record of the past few seconds is enough to predict a person's activity. This is the basis of the sliding window concept.

For example, when a person runs, the component of acceleration perpendicular to the floor will change sign rapidly in a rhythmic fashion. If we can detect the presence of such a rhythmic pattern and perhaps also obtain the frequency of this pattern, it could be a useful feature to detect running. In contrast, a single observation will probably be useless, since there will be so much variation in individual samples.

A window is defined as a contiguous subsequence of samples. Samples numbered from $ti - t + 1$ to $ti - t + k$ make up the i^{th} window of size k. t is a parameter which decides the amount of overlapping between windows.

In this approach, features are extracted from readings in each window. Extracted features generally have some meaning associated with them. For example, the Fourier transform can help us get the frequency of a rhythmic pattern. We can often extract features characteristic to certain activities and these features can help us differentiate between activities.

People in this dataset don't change their activities rapidly and sampling frequency is 52 Hz. Therefore, for any given activity, we have thousands of consecutive samples. Any window of 10 to 100 samples is very likely to have the same activity.

Features previously extracted [1] from accelerometer readings include:

- mean for each component
- standard deviation for each component
- correlation between components
- RMS velocity for each component (obtained by numerically integrating acceleration)
- Range of acceleration for each component (maximum value minus minimum value)

In addition, instead of using only the original 3 components, they have also used:

- Total magnitude of acceleration, obtained by taking L2-norm of all components.
- A band-pass-filtered signal instead of the original signal.

After features have been extracted from a sliding window, we can train a classifier on these features. Casale et al obtained classification accuracy of 94% using Random Forests. [1]

We trained random forests with 10 and 30 trees. We also trained Linear SVM and Logistic Regression. All of them gave accuracy between 92% and 94%.

3 Data Analysis and Preparation

3.1 Relabeling

Some classes in the data are very similar. For example, working on a computer and standing are very similar activities. Both involve a person being stationary, so they will give the same accelerometer readings. Activities like 'standing' and 'standing while talking' are also similar, because talking generally doesn't affect a person's movement. Therefore, we merged such similar classes.

This is the mapping we used:

- (1) Working at computer → (0) Stationary
- (3) Standing → (0) Stationary
- (4) Walking → (1) Walking
- (5) Going up-down Stairs → (2) Stairs
- (6) Walking and talking with someone → (1) Walking
- (7) Talking while standing → (0) Stationary

We dropped the class '2 - Standing up, Walking and going up-down stairs' since it wasn't clear what that meant.

3.2 Problems with Data

We have data of just 15 participants, which is very less. However, for each participant, we also have many (around 100k) accelerometer readings.

Class distribution:

- Stationary - 75%
- Walking - 22%
- Stairs - 2.7%

The participants spend very less time climbing stairs (probably because it's tiring). Therefore, there is a big class imbalance against climbing stairs.

3.3 Analysis of Plots

Plotting data with time and coloring the plot by activity type suggests that walking and stairs cause a lot of accelerometer fluctuation.

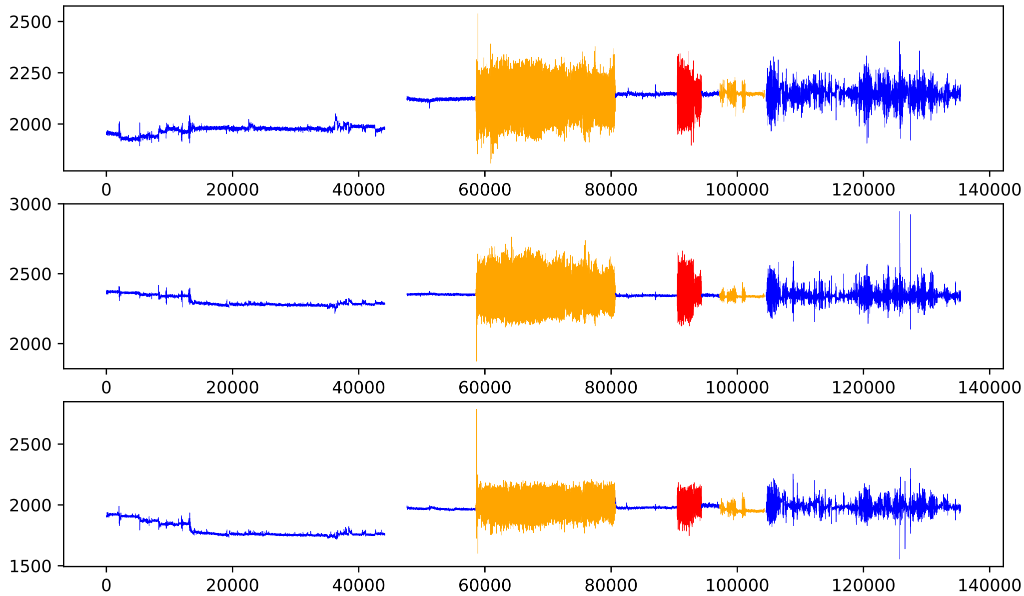
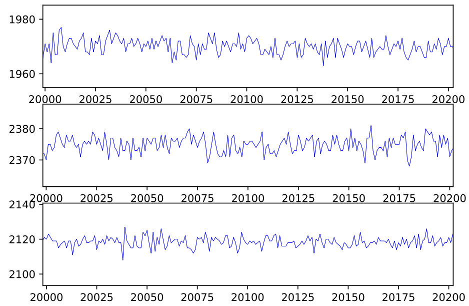


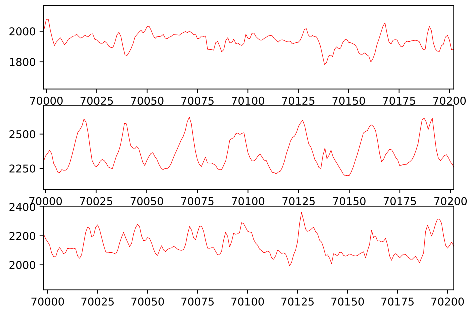
Figure 1: Accelerometer readings by label
(blue - stationary, orange - walking, red - stairs)

Looking at the plot at a microscopic level, we can see that walking and stairs generate accelerometer readings with frequency around 50 Hz in the x axis and 25 Hz in the y and z axis. Being stationary doesn't generate readings of a discernible frequency.



(a) Stationary

(b) Walking



(c) Stairs

Figure 2: A closer look (around 4 seconds) at different activities.

Applying Discrete Fourier Transform (DFT) didn't generate a pattern which indicated the presence of particular frequencies. All frequencies seemed to contribute equally. Hence, we couldn't use DFT to distinguish patterns in the frequency domain.

3.4 Feature creation

We calculated the magnitude of acceleration as the L2-norm of all components ($a_m = \sqrt{a_x^2 + a_y^2 + a_z^2}$).

We couldn't find ways to band-pass-filter readings.

So we now have 4 series: a_x, a_y, a_z, a_m.

3.5 Windowing

Data is split into windows of size 100, where consecutive windows are 5 observations apart ($k = 100, t = 5$).

For each window and each series, we calculated the following statistics:

- Standard deviation
- Range (maximum minus minimum)
- Skewness

We didn't use mean or median since accelerometer readings are uncalibrated, so mean and median varied from participant to participant.

For each window, label is set to the label of the last sample. This choice doesn't matter much, since participants don't rapidly change their activities, so all samples in almost all windows will have the same label.

We concatenated all windows of every participant to get a matrix with 12 columns.

Since Logistic Regression and Linear SVM are scale sensitive, we scaled all features by z-normalization (subtracting mean and dividing by standard deviation).

4 Results and Discussion

We used 4 classifiers:

- Random Forest with 10 trees (RF10)
- Random Forest with 30 trees (RF30)
- Logistic Regression (LogReg)
- Linear SVM (LinSVM)

Training and evaluation was done using 5-fold cross validation.

Table 1: Training Accuracy

	LogReg	LinSVM	RF10	RF30
mean	94.0%	93.7%	99.85%	99.983%
stddev	0.92%	0.85%	0.03%	0.003%

Table 2: Testing Accuracy

	LogReg	LinSVM	RF10	RF30
mean	93.5%	93.6%	92.0%	92.1%
stddev	3.4%	3.3%	3.5%	3.5%

It can be concluded that all the above classifiers have comparable accuracy.

Since 'Stairs' has scarce representation, it is likely to be misclassified without causing much decrease in accuracy. We calculated its precision and recall on both training and testing data.

Table 3: Precision and Recall for 'Stairs' on Training data

	Mean Precision	Mean Recall
LogReg	10.50%	0.16%
LinSVM	7.11%	1.90%
RF10	99.96%	98.53%
RF30	100.00%	99.85%

Table 4: Precision and Recall for 'Stairs' on Test data

	Mean Precision	Mean Recall
LogReg	5.81%	0.70%
LinSVM	6.61%	1.59%
RF10	23.19%	18.37%
RF30	21.72%	17.60%

Since precision and recall are low for Logistic Regression and Linear SVM in both training and test data, it indicates that 'Stairs' is not linearly separable from other classes.

4.1 Feature Importances

For a decision tree, the importance of a feature is the sum of reductions in Gini value in every node it appears. For a random forest, the importance of a feature is the sum of its importances across all trees.

Table 5: Feature importances

	Mean RF10	Mean RF30
x_{sd}	3.34%	2.19%
x_{range}	5.03%	3.04%
x_{skew}	2.28%	3.51%
y_{sd}	24.68%	24.78%
y_{range}	15.48%	10.90%
y_{skew}	15.65%	17.97%
z_{sd}	2.33%	2.76%
z_{range}	2.41%	3.20%
z_{skew}	2.51%	2.94%
m_{sd}	10.93%	9.73%
m_{range}	5.45%	7.75%
m_{skew}	9.91%	11.23%

This is in line with our expectations, since activities like walking and climbing stairs involve a lot of up-down movement of a person (i.e. in the y-axis) and very small sideways movement.

5 Conclusion

Random forests, Logistic Regression and Linear SVM are almost equally capable at predicting human activity type from accelerometer readings.

All classifiers give poor precision and recall for the 'Stairs' class. Random forests perform poorly probably due to limited data in the 'Stairs' class. Logistic Regression and Linear SVM perform poorly due to 'Stairs' not being linearly separable from other classes.

We find that the direction perpendicular to gravity is the most important for predicting human activity, based on feature importances given by random forests.

6 Applications and Scope

Wearable computing platforms can be used to monitor day-to-day activities and be used as personal digital assistants. A system being aware of both context and activities during daily life would not just be able to give assistance in memory-retrieval tasks, but also for real-time assistance to not completely self-sufficient people.

In many recent works activity recognition is based on classifying sensory data using one or many accelerometers. Accelerometers have been widely accepted due to their compact size, their low-power requirement, low cost, non-intrusiveness and capacity to provide data directly related to the motion of people.

Biometric activity patterns can also be used for personalization and user identification. However, we consider that to be beyond the scope for this assignment and we will only focus on activity recognition.

Wrist-mounted systems are generally deemed better because they are more convenient for the user. However, chest-mounted systems should nevertheless be explored since they give different sensory measurements and so they might be more suitable in certain scenarios or for certain purposes.

References

[1] Pierluigi Casale, Oriol Pujol, and Petia Radeva. Human activity recognition from accelerometer data using a wearable device. In *Proceedings of the 5th Iberian Conference on Pattern Recognition and Image Analysis*, IbPRIA'11, pages 289–296, Berlin, Heidelberg, 2011. Springer-Verlag.

[2] Christopher Olah. Understanding lstm networks - http://colah.github.io/posts/2015-08-understanding-lstms/, 2015.