Introduction

Methods

Aims

The overarching aim of this project is the accurate prediction of activity label given accelerometer and gyroscope sensor data. To achieve this aim several models shall be implemented and trained upon a selected multi-person dataset.

Programming and computation

Computational work was conducted using R version 3.4.4 via the IDE software suite R studio. Any packages and functions used in this project will mentioned where appropriate. Code and scripts can be found on GitHub () exactly as they were used for reproducibility purposes along with data output and initial dataset.

Dataset

The dataset used for this project was acquired from the UCI Machine Learning Repository site, <https://archive.ics.uci.edu/ml/index.php>, with the specific dataset being found at [https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones#](https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones). This data was sourced from the work of Anguita *et al* (2013) based on human activity recognition using smartphones. This dataset was created using 30 individuals with an age range of 19-48 years performing 6 activities whilst wearing a Samsung Galaxy S II smartphone. Data was recorded using the embedded accelerometer and gyroscope sensors capturing linear acceleration and angular velocity in three axes at a constant frequency of 50Hz. Data was then labelled manually according to visual data and was subsequently partitioned at random into a 70-30 train-test split (Anguita *et al*., 2013).

Pre-processing:

Data was pre-processed by the authors, this involved application of noise filters with subsequent sampling by fixed width windows of 2.56 seconds with a 50% overlap resulting in 128 readings per window. The accelerometer signal consisted of gravitational and body motion components which were separated using a Butterworth low-pass filter into body acceleration and gravity respectively. They expected gravitational force to have low frequency components only, as such a filter with a 0.3Hz cut-off frequency was applied. A vector of features was taken from each window by calculating variables from the time and frequency domains (Anguita *et al*,. (2013)).

Feature selection

The dataset originally possessed 561 attributes not including the labels, activity, and ID attributes that were added for the purpose of training. This results in a high degree of dimensionality for this dataset which both causes troubles for machine learning algorithms and increases computational expense. As such an extensive feature selection workflow was implemented, this work flow contained three steps and parameters were altered to discover optima. Firstly feature correlation was calculated in the form of a correlation matrix, this was assessed to find features with an absolute correlation above a specified cut-off. Two sets were produced from this step, one with a cut-off of 0.5 and the other a cut-off 0.75 absolute correlation. The next step involves implementing cross validation in conjunction with two models; a learning vector quantization model and a random forest implementation. Importance was then quantified for each feature using various functions found in the caret package and randomForest package. Functions, models, and parameters will be further explained in the model’s section.

Figure n. Lists of feature importance produce by A) the importance() function from the randomForest package and B) the varImp function from the caret package showing MeanDecreaseGNI and Overall importance scores respectively.

Distribution of data –

The distribution of the data was identified with the descdist () function from the fitdistrplus package. This function was applied to each feature in the dataset to estimate distributions, the Cullen and Frey graphs for the 5 most important features output by this function can be seen in Figure n. The data was found to be best fit by the Beta distribution and this property was noted during model selection.

Model selection –

Four models were chosen and implemented for predictive modelling of activity labels; a Random Forest, a Naïve Bayes model, a gradient boosting model, and a decision tree. Reasoning behind each selection is explained for each model respectively in the model’s section.

Models

Learning Vector Quantization (LVQ);

This algorithm is important in statistical pattern classification, the basis behind the function of this model is building a quantized approximate of input data distribution, this is done using finite numbers of a prototype, or vectors. Prototypes in LVQ result from a procedural update based upon training data, once set an assortment can be associated with segments or classes within the data, they can also be employed to classify datapoints via nearest neighbour rules (

This model was implemented using the caret package train() function with 10-fold repeated cross-validation repeated 5 times as a control, scaling was applied in the parameters. The varImp() function from the caret package was subsequently applied to quantify importance the output of which can be seen in Figure nA and Figure NX. Selection of this model for use in feature selection was based upon its use in several literature examples; Prezenger *et al* (1996), Flotzinger *et al* (1994), and Suutala and Roning (2004) to name a few. This method was used to great effect in these literary examples and is recommended for automated feature selection which was subsequently implemented in this study.

Random Forest (RF):

A random forest model was used in conjunction with the same cross-validation, this model was chosen for feature selection based upon literary implementations in the same vein. Notably Chen and Liu (2005) evaluated this method of feature selection in conjunction with an SVM where it performed well in comparison to other similar methods. Tree sizes were set between 1 and 85 (the number of features in the train set) for this implementation.

Another random forest model was implemented during the modelling stage of this project, this was chosen based on its effectiveness in predictive modelling shown in literature such as Shaikhina *et al* (2017). This was implemented with the train () function from the caret package much like the other models implemented in this project. Pre-processing like the others was set to centre and scale, with the results of testing predicted using the cofusionMatrix () function.

Subspace clustering (SC):

A predictive subspace clustering model was chosen and implemented based upon properties of the target dataset. It was noted earlier on that the dataset suffers from high dimensionality and whilst this issue was dealt with in part by the feature selection process the resulting subsection of data still exhibited this property to a much lesser degree. It was chosen to include at least one clustering model to ensure a diverse selection of models for later comparison. However cluster detection has issues with high dimensional data, to be specific as dimensionality increases the distance between all points in the dataset becomes equal due to only having a finite number of points in an increasingly larger space (Steinbach *et al*., 2011). Subspace clustering can deal with this problem by, simply put, taking subsections of the dataspace and clustering each independently effectively dealing with lower dimensional spaces. In a predictive context subspace clustering was implemented effectively by McWilliams and Montana (2013), the model in this project was in part based on literary examples such as this and adapted to fit the target problem. This implementation was conducted using the CLIQUE algorithm found within the subspace package.

Naïve Bayes classifier (NB):

This is a form of statistical classifier able to predict class membership probabilities and then choose the most likely class based upon this. This method was chosen due to both its ability to handle multiclass classification as well as its performance in comparison to other classification models as shown by Ng and Jordan (2002). In comparison to some models Naïve Bayes is relatively simple but it may also be very accurate if correctly implemented. This model l like the others was implemented using the train () function from the caret package with parameters left as default and the model testing was performed using the predict() function with visualization by the confusionMatrix () function.

Gradient boosting model (GBM):

A gradient boosted regression model was implemented for predictive modelling, such methods are often included in comparative

Decision tree (DT):

A comparative analysis, Tso and Yau (2006), showed decision tree models as a competitive option for predictive modelling and based on this it was decided to include it in this project. Like the rest of the models presented for predictive modelling so far this was implemented via the caret package with the train () function with preprocessing again set to centre and scale. A confusion matrix was visualised for the output using the predict () and confusionMatrix() functions.

Results

Discussion

References

Chen, Y. and Lin, C. (2018). Combining SVMs with Various Feature Selection Strategies. *Feature Extraction*, [online] pp.315-324. Available at: https://link.springer.com/chapter/10.1007/978-3-540-35488-8\_13 [Accessed 24 Jun. 2018].

McWilliams, B. and Montana, G. (2013). Subspace clustering of high-dimensional data: a predictive approach. *Data Mining and Knowledge Discovery*, 28(3), pp.736-772.

Ng, A. and Jordan, M. (2018). On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes. [online] Available at: https://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf [Accessed 24 Jun. 2018].

Pregenzer, M., Pfurtscheller, G. and Flotzinger, D. (1996). Automated feature selection with a distinction sensitive learning vector quantizer. *Neurocomputing*, 11(1), pp.19-29.

Pregenzer, M., Pfurtscheller, G. and Flotzinger, D. (1996). Automated feature selection with a distinction sensitive learning vector quantizer. *Neurocomputing*, 11(1), pp.19-29.

Shaikhina, T., Lowe, D., Daga, S., Briggs, D., Higgins, R. and Khovanova, N. (2017). Decision tree and random forest models for outcome prediction in antibody incompatible kidney transplantation. *Biomedical Signal Processing and Control*.

Steinbach, M., Ertoz, L. and Kumar, V. (2018). *The Challenges of Clustering High Dimensional Data*. [online] Www-users.cs.umn.edu. Available at: https://www-users.cs.umn.edu/~kumar001/papers/high\_dim\_clustering\_19.pdf [Accessed 23 Jun. 2018].

Suutala, J. and Roning, J. (2004). TOWARDS THE ADAPTIVE IDENTIFICATION OF WALKERS: AUTOMATED FEATURE SELECTION OF FOOTSTEPS USING DISTINCTION-SENSITIVE LVQ. [online] Available at: http://www.ee.oulu.fi/research/bisg/files/pdf/pdf\_515.pdf [Accessed 24 Jun. 2018].

Tso, G. and Yau, K. (2007). Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. *Energy*, 32(9), pp.1761-1768.

Yi-Wei, C. (2005). *Combining SVMs with Various Feature Selection Strategies*. pp.pp 315 - 324.

Zhang, Y. and Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, 58, pp.308-324.