Action recognition using accelerometer data

Assignment 3
Image Based Biometrics 2019/20, Faculty of Computer and Information Science, University of Ljubljana

Kristjan Reba

***Abstract*—Human action recognition (HAR) has benefited from large amounts of data, increase in computing power and improved machine learning methodology. We look at the problem of classifying human activity based on time series data obtained by accelerometer sensor. We evaluated different machine learning algorithms such as Support vector classifier (SVC), multi-layer perceptron (MLP), neural network using LSTM cells, K nearest neighbours (KNN) and Light Gradient Boosting Machine (LGBM). We show that simpler models like KNN can outperform more complicated models like deep neural networks. We also test the window size for LSTM model and conclude that window size is positively correlated with the accuracy, but it succumbs to diminishing returns. The best performing model is KNN with the mean accuracy on 5-fold cross validation being 99.6%.**

I. Introduction

Human action recognition (HAR) is an important part of many applications. It can be used for gathering valuable statistics about human behaviour [1] or just for tracking personal fitness through smart devices. In recent years there has been large progress made in HAR due to large increases of data and computing power which enabled more complex models such as deep neural networks to come into consideration [2], [3]. In this paper we are interested in comparing different classifiers for HAR and how well they perform. We perform the evaluation on time-series data recorded by an accelerometer sensor attached to a person who performs different activities. We want to create a reliable and robust classifier that can be used for HAR and many subsequent applications.

II. Methodology

For our purposes we evaluate three different classification models. We use support vector classifier (SVC), multi-layer perceptron (MLP) and long short term memory deep neural network (LSTM). The dataset we use consists of 102422 data samples. Each sample is a recording accelerometer sensor attached to a person performing an activity. For evaluation we run a 5-fold cross validation (CV) to get a good estimate of models quality. We choose to shuffle the dataset to get a representative sample of classes in each fold. Using original dataset gives low accuracy due to unseen classes during training. Because of unbalanced class distribution, presented in figure 1, we choose to perform oversampling on the dataset. For the oversampling algorithm we choose synthetic minority oversampling technique (SMOTE) [4]. In the following subsections we briefly describe each of the models we use.

A. Support vector classifier (SVC)

One of the algorithms we test is support vector classifier which is a just a support vector machine (SVM) adjusted for classification tasks [5]. SVMs have shown great predictive power on various tasks and were only recently surpassed by deep neural networks (DNN) in their performance. For our experiments we choose to use the *rbf* kernel and have tested various values of regularization parameter.

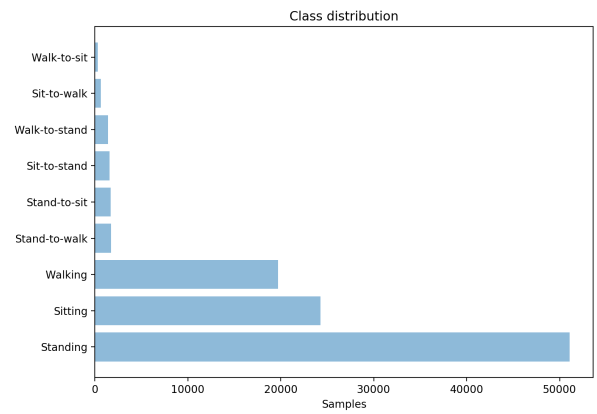

Figure 1. Class distribution

B. Multi-layer perceptron (MLP)

Classic fully connected feed forward neural network also referred to as multi-layer perceptron is another extremely powerful function approximator that we evaluate on presented HAR task [3], [6]. We use the MLP with hidden layer sizes of 32, 32 and 16 in that order. For optimization of the neural network we use the Adam optimizer [7].

C. Neural network with long short term memory cells (LSTM)

To make use of sequential nature of the data we decide to use a deep neural network consisting of long short term memory (LSTM) cells [8]. We evaluate different window sizes, meaning different number of data samples recorded right before the current data sample. The LSTM neural network we use 3 hidden layers of LSTM cells with sizes 32, 32 and 16 in that order.

D. Light Gradient Boosting Machine (LGBM)

Light Gradient Boosting Machine is an efficient and scalable version of Gradient Boosting Decision Tree (GBDT) [9]. To achieve better efficiency the authors of the algorithm present two ideas: dataset sampling based on gradient and merging the features into bundles. This modifications allow LGBM to achieve faster training speed and higher data efficiency as well as lower memory usage than other boosting algorithms.

E. K-Nearest Neighbours (KNN)

In our experiments we use a KNN algorithm with K equal to 1. The algorithm stores each data samples from the training set and does not do more learning. KNN makes predictions based on the K closest neighbours to the given data sample we want to classify. If K is set to 1, then the algorithm will classify any test sample in the same class that the closest training sample is in.

Table I
Results table contain the accuracy of each model on 5-fold cross validation.

Majority class	SVC	MLP	LSTM	LGBM	KNN
0.498	0.934	0.946	0.970	0.993	**0.996**

III. Results

For the baseline model on chosen dataset we decided on majority class classifier which achieves the accuracy of 49.8%. In the table I we present the results of our experiments and accuracy the models achieved on stratified 5-fold cross validation using the whole available dataset. The data is shuffled before splitting to obtain a representative sample in the training set as well as in the test set. Because the standard deviation of achieved results is extremely small (less then 0.001) we choose to omit it and only present the mean accuracy for each model. We observe that LSTM model improves over simple MLP with margin over 2%, while SVC fails to reach the accuracy of MLP. The best performing models are Light Gradient Boosting Machine and a simple K-Nearest Neighbours with K equal to 1. Both algorithms achieve a score higher than 99% with KNN being the best and achieving the accuracy score of 99.6%.

We also test different window sizes for the LSTM model and see that accuracy increases with the number of previous data samples we merge into a single data sample. Figure 2 presents the accuracy for each tested window size. We see that the highest increase in accuracy comes from the first 32 samples and latter succumbs to diminishing returns, where we increase window size and only get a minor improvement in accuracy.

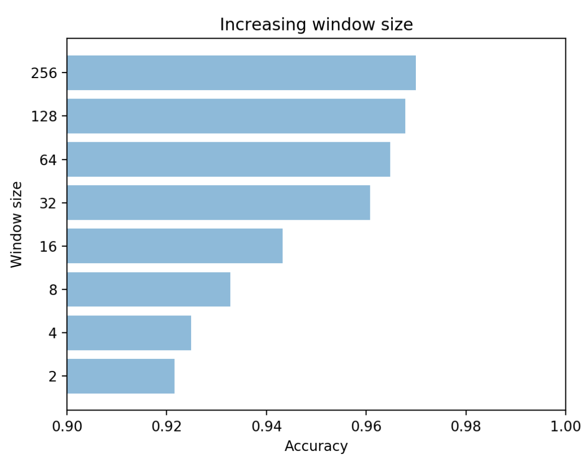

Figure 2. Increasing window size for LSTM model.

IV. Conclusion

Human activity recognition from accelerometer data is an important task for many subsequent applications. We evaluated the performance different algorithms on time series data obtained by accelerometer sensor attached to a person. We conclude that making use of time dependence of samples can increase the performance of the models, but the best performing algorithm is a simple KNN which makes predictions only based on the closest data sample from the training set. We also showed the importance of window size for the final accuracy of the LSTM model. For further work it would be interesting to increase the complexity of the models by including the Bidirectional LSTM (BiLSTM). We could also try some prepossessing techniques and data augmentation techniques such as adding random noise and different types of normalization. One possibility would be building an ensemble model out of the best performing models and trying to improve on the KNN accuracy score.

References

[1] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, “Deep learning for sensor-based activity recognition: A survey,” *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[2] N. Twomey, T. Diethe, X. Fafoutis, A. Elsts, R. McConville, P. Flach, and I. Craddock, “A comprehensive study of activity recognition using accelerometers,” in *Informatics*, vol. 5, no. 2. Multidisciplinary Digital Publishing Institute, 2018, p. 27.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[5] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[7] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[8] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” in *Advances in Neural Information Processing Systems*, 2017, pp. 3146–3154.