ML4QS: Assignment3 - Group 73

Tianhao Xu[1 [2740825]], Han Lin[2,3 [2724786]], and Yiyang Qi[3 [2721018]]

Vrije Universiteit Amsterdam

1 Introduction

Nowadays, more and more working people are lacking of adequate physical activity due to the pressure of their work. To address this problem, experts are researching physical activity monitor (PAM) to quantify physical activity through exercise data to improve people's health habit.

The aim of our work is to predict the activity performed by the user at each measurement through a machine learning approach. We plan to use IMU sensors to track the daily life of 9 subjects. Our experimental data is based on the UCI's PAMAP2 Physical Activity Monitoring Data Set [1], with a total of 18 activities. The individual's self is quantified by temperature, acceleration, gyroscope and magnetometer.

An exploratory analysis of the data is given in Section 2, followed by Data Pre-processing and Feature engineering in Sections 3 and 4, respectively. Section 5 provides the evaluation of relevant methods. Section 6 gives a detailed description and discussion of the results. Finally, Section 7 includes conclusions and directions for future work.

2 Exploratory Data Analysis

The dataset consists of nine .bat files, each corresponding to test data from a subject performing 18 activities(all activities last approximately 3 minutes). The raw data was collected by 3 IMU sensors ana a heat rate monitor. Sensors are located on the wrist of the dominant arm, on the chest and on the ankle, and such sensors are sampled at a frequency of 100 Hz while that of HR-monitor is 9 Hz. 12 different activities were performed in each subject (lying, sitting, standing, walking, running, cycling, Nordic walking, vacuuming cleaning, ironing, rope running, ascending and descending stairs), in addition to six different activities of daily living (watching TV, computer work, car driving, folding laundry, house cleaning, playing soccer). The total duration of all activities is over 10 hours.

The dataset has 2872533 instances and 55 features, each instance stands for an user's activity data during experiment. The columns include subjectID, timestamp(s), activityID, heart rate(bpm), hand temperature, 3D-acceleration data (ms^{-2})(scale: ±16g), 3D-acceleration data (ms^{-2})(scale: ±6g), 3D-gyroscope data (rad/s), 3D-magnetometer data (μT) and orientation (invalid in this data collection). As different parts of the body collect the same data content, we present here only the hand data, and the other parts in turn.

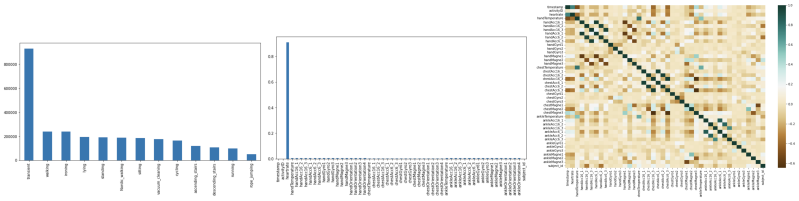


Fig. 1. Distribution of ActivityID

Fig. 2. Percentage of Missing Values

Fig. 3. Correlation Matrix of Features

Fig.1 shows the distribution of the labels in the dataset. As can be seen, the most frequent occurrence is "transient", while the distribution of other actions is less distinct. This type of data will be removed in data pre-processing, as it is a transient activity between different activities and has no practical significance for analysis. Also missing from the raw data are some activities, including watching TV, computer work, playing soccer, folding laundry, house cleaning and car driving. The data is indeed probably due to wireless sensors not all being sampled at the same moment or hardware connection interruptions[2]. Considering the large number of labels present, the different percentage of labels would not lead to a problematic class imbalance. And as the above plot shows, useful classes are mostly balanced.

It is obvious that the missing values widespread in various features, as shown in Fig.2. The percentage of missing heart rate is greater than 0.9, while the percentage of missing other data is less than 0.005, and this is due to the different frequency of instrument collection.

We checked the correlation between the features and we removed the irrelevant features mentioned in the document and the correlation coefficient results are shown in the Fig.3. Overall, there is a high correlation between acc16 and acc6 for each body part, chest_acc16_3 and chest_acc6_3 have a more significant negative correlation with activityID.

In order to calculate different granularities, we split the original data set into 6 separate datasets of acceleration, gyroscope, and magnetic force according to the hand monitor and the chest detector. Fig.4 and Fig.5 show recorded values of the different attributes with granularity 60s and 0.25s respectively. The described effect can be observed in these figures as the high granularity data is more smoothed than the low granularity data. On the other hand, the low granularity data contains large variability. In the Fig.4 and Fig.5, the gyroscope does not show remarkable observations. The gyroscope sensor indicates that the phone was kept in the same orientation for most passive activities, like sitting. Acceleration showed indirect small-amplitude fluctuations, indicating that subjects have motion state change. In addition, Fig.6 and Fig.7 shows the accelerometer amplitude is similar for the two granularity, but in the Fig.6, it can be seen that the spread of the Acceleration is smaller than the spread seen in Fig.7. Based on the insights we have just gained, we would lose too much information and valuable training data if we were to use the coarse-grained variant, thus we finally select the most fine grained dataset(Δt=250 ms)[3].

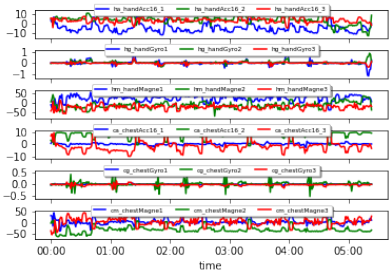


Fig. 4. Measured signals at Δt=60 s

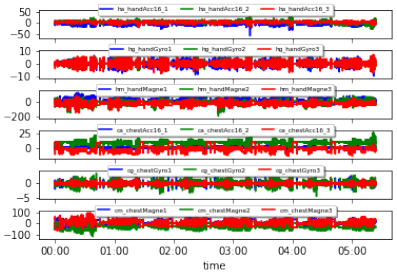


Fig. 5. Measured signals at Δt=250 ms

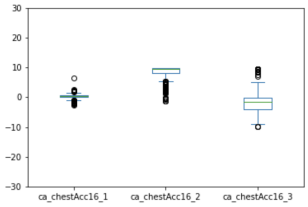


Fig. 6. Boxplot of chest accelerometer data with Δt=60s

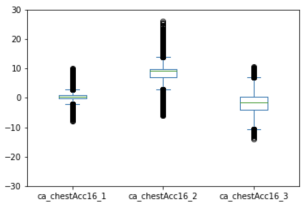


Fig. 7. Boxplot of chest accelerometer data with Δt=250 ms

3 Data Pre-processing

3.1 Missing Values

Firstly, as Fig.2 mentioned, 53 features contained missing values, with heart rate being as high as 90%. To check if heart rate is indeed needed, we plotted the average heart rate frequency for the different activities. As shown in Fig.8, running and rope_jumping are the most strenuous, and 9Hz is still sufficient to capture all the variability in the IMU data (acceleration oscillations and angular velocity oscillations). Therefore, we first removed the data without heart rate, i.e. the time step is closer to 0.1 seconds than the time step of 0.01[4].

Since activityID 0 denotes a temporary state, it should be fully dropped from the dataset (as mentioned before). Additionally, we eliminated the orientation of the three body parts(12 features total) since it seemed to be pointless and would require extra training time for the model[2]. 43 basic features remain at this step.

Each activity contains NaN values since it only has 8 or fewer subjects. Using interpolation is the simplest method for getting rid of NaN values. We first removed the non-numeric data and dropped cells with NaN value by constructing new data points from this known set of data points. Since the heart rate feature's initial cells were NaN and interpolation calculates values around the NaN cell, this feature still contained NaN. We assumed that the value of the first 4 cells is 100 to solve this issue, since the heart rate value after the index 4 is 100. Any NaN values are removed from our dataset at this stage.

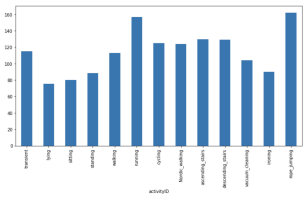


Fig. 8. Avg Heart Rate by Activity

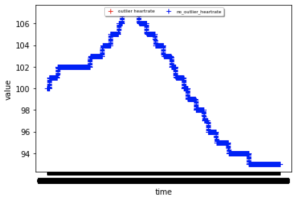


Fig. 9. Outliers with Simple distance-based approach(dmin=0.15, fmin=0.99)

3.2 Outliers

We concentrated on heart rate since, as suggested by the different indicators in the self-reporting file for the dataset, it is the most precise gauge we have for monitoring people during the event. Therefore, we examined heart rate potential outliers. Since Chauvenet's criterion assumes a normal distribution of the data[5], while heart rate is not normally distributed. We calculated the average heart rate corresponding to the different activities in Fig.8. With the exception of running and jumping, there was little variation in heart rate and no significant segmentation, so it did not fit into the Mixture Models either. Due to its high computing cost, the LOF algorithm is not used for our outlier detection[6]. Therefore, we applied the Euclidean Distance and due to memory issues, we only show the results for the first 5000 data items, as shown in the Fig.9, and there are no obvious outliers. We verified this conclusion using the other three methods as well, and subsequently checked the other features and there are no outliers.

3.3 Split Dataset

We ended up total of 1942872 out of the previous 2872533 instances with 42 basic attributes and a target. The dataset is split randomly into a training set (80% of the data, 1554298 instances) and a test set (20%, 388574 instances).

4 Feature Engineering

4.1 Time-based

We found that the information obtained in the original signal was not obvious. Therefore, more information can be obtained by extracting statistical features using the time domain. For basic features, three window sizes that 30, 60 and 120 measurements are used for the time-based feature engineering. Given the frequency of 30 measurements per second, this corresponds to a window size of 1.1, 2.35 and 4.29 seconds respectively. We obtained 42*2 = 84 time-based features using the mean and standard deviation of basic features.

4.2 Frequency-based

We also used frequency domain analysis, which can be used to calculate the spectral information of the signal, in the hope of extracting more reliable features. 15 measurements that exhibited periodicity are selected from the frequency domain. 3 aggregation functions are applied in the frequency domain with 18 different frequency amplitudes as well. Considering that multiple window sizes are used, the overlapping implementation uses 30 window sizes for the measurements. This totals to $15 * (3 + 18) = 315$.

4.3 Principal Component Analysis (PCA)

When performing training and testing, the model may take lots of time to execute due to the high number of features. However, we can predict and reduce the number of features with the help of PCA[11] to reduce the corresponding time. PCA in sklearn has two ways of initializing objects that specifying the number of principal components and specifying the amount of retained variance.[12] We used the approach of specifying the number of principal components by setting the components as 17 and adding them to the training and test sets.

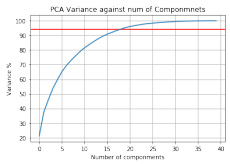


Fig. 10. PCA Variance against number of Componments

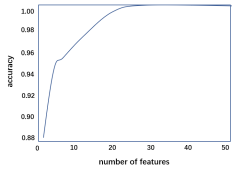


Fig. 11. Feature selection for the label prediction task (training set)

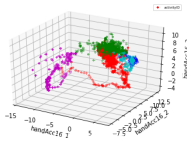
Typically, 90%-98% of the variance explains the data well. As Fig.10, 17 components account for approximately 94% of the variance.

4.4 Feature selection

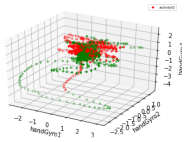
In total, we have 42+84+315+17=458 features. Then We performed the forward selection to get important features because useless features can affect the performance of the algorithm. We used a decision tree classifier to predict the accuracy of the features and to determine which features would improve our performance. We iteratively added the impact of the best features to the first 50 features as Fig.11. The final 25 features with high accuracy are selected. Top 3 features are $'handAcc16\_1\_temp\_std\_ws\_30'(1.0), 'handAcc16\_3\_temp\_mean\_ws\_30'(0.999), 'cankleAcc16_2\_temp\_std\_ws\_30'(0.996)$.

4.5 Clustering

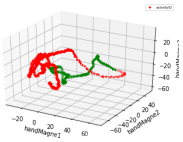
We referred to the method of the book[13], and choosen k-medoids to cluster Acceleration data, Gyroscope data and Magnetometer data in three sensors IMU hand and IMU ankle respectively. We used the different activityID as the label. K-medoids uses medoids as reference points and is very good at weakening the



(a) Plot of Acceleration data clustering with IMU hand, k = 5



(b) Plot of Gyroscope data clustering with IMU hand, k = 2

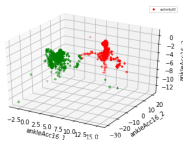


(c) Plot of Magnetometer data clustering with IMU hand, k = 3

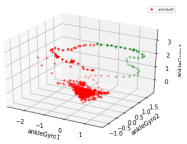
Fig. 12. The clustering result of IMU hand

effect of outliers. Through the 3D distribution of k-medoids, we want to see whether results of clustering are the same for different sensors with the same data and for different data of the same sensor.

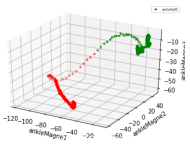
By plotting the relationship between silhouette coefficients and k values(2-9), we chose the highest silhouette coefficients and drew the clusters of the two sensors (IMU hand in Fig.12, IMU ankle in Fig.13). We found that the clustering of the same data was different for different sensors. In the case of Acceleration data, for example, the best clustering is found for the IMU hand sensor, k=5. It can be found that there are five clusters as Fig.12(a).



(a) Plot of Acceleration data clustering with IMU ankle, k = 2



(b) Plot of Gyroscope data clustering with IMU ankle, k = 2



(c) Plot of Magnetometer data clustering with IMU ankle, k = 2

Fig. 13. The clustering result of IMU ankle

Also, the clustering is different for various data within the same sensor. Taking the IMU ankle as an example, the Acceleration data clustered best at k=2, with two clusters on either side in Fig.13(a).

5 Evaluation

We tried to use self-supervised task for classification and prediction for a given time and values of the sensor activities. Since our data set is in large data volume which needs ability to learn high-order features, we need nonlinear models used to fit the data to gain accurate results. Therefore, we considered creating a benchmark using DTR as well as building NN model using LSTM.

Accuracy, Recall, Precision, and F1 score are four regularly used classification metrics that are used to assess models in a multi-classification issue. Accuracy as the main metrics. Results of each model are shown in Table 1 with its multiple feature sets, including the basic feature set (42,'b') and the final feature set (25,'f'). For final model we realized comparison experiments on the basic feature set and the selected feature set, and adopted 10-fold cross-validation.

Model	Accuracy		Recall		Precision		F1-score	
	Training	Test	Training	Test	Training	Test	Training	Test
DTR(f)	93.14%	56.80%	91.40%	56.27%	92.12%	55.28%	91.74%	55.93%
LSTM(f)	89.87%	34.06%	89.65%	33.17%	79.08%	34.84%	89.02%	33.64%
Optimised LSTM(b)	98.27%	92.55%	98.94%	91.83%	99.75%	92.83%	99.74%	91.74%
Optimised LSTM(f)	99.99%	99.99%	99.99%	99.99%	99.99%	99.99%	99.99%	99.99%

Table 1. Performance of algorithms

DecisionTreeRegressor(DTR) Decision tree is one of the well known and powerful supervised machine learning algorithms that can be used for classification and regression problems. The model is based on decision rules extracted from the training data. We used the default parameters of the DTR class. The default values are shown as following: criterion='mse', min_samples_leaf=1, min_samples_split=2, presort='deprecated', random_state=None, splitter='best'.
LSTM Considering the complexity of our data set, we used LSTM to deal our time series which has better performance in longer sequences compared with normal RNNs. First we preprocessed the data using a sliding window and set the parameters step_back=5, step_forword=1 to predict from past 5 times activities to 1 next time activity. We employed the *keras* library and added two layers in the sequential container and set the dense to 10, and specified the number of epochs training rounds as 10 and the validation_split as 0.2. But the accuracy score we got is only 34.06% on the test, this is even a lower result than DTR.
Final model (Optimized LSTM) Since the LSTM results are not ideal, especially performance on the test set, whose accuracy is only 34.06%. So we tried to optimise it. Firstly, We added 4 more dense layers in the LSTM model to improve the performance. In these layers, the *units* equal 64 and set *relu* as the activation function to allow the Neural network to learn nonlinear dependencies. Secondly, the DTR and LSTM models all exhibit superior performance on the training set compared to the test set, as can be shown in Tab.1. It indicates that it would be due to overfitting, which could not be diminished during model adjustment. Therefore, we applied dropout algorithm starting with 0.1, optimized

these features. The accuracy score of LSTM after optimization reached 99.99% on test set. For cross validation, optimized LSTM is our final model, the mean value of val_accuracy for final features is 99.99%, compared with basic features is 94.61%.

6 Result

After optimizing features and layers appropriately would give the model a higher ability to reduce the loss of mean square error in the training data, which results in improved performance of the model.
It can also be noted that the features produced by feature selection typically performs better than the features from the original dataset. The optimized LSTM model appears to do the best, according to results. The accuracy of this model, which makes use of final features, are both 99.99% on the training set and test set. Fig.14 depicts the confusion matrix for this model. It is clear that our predictions work very well and are almost always correct. It can be seen from the confusion matrix that only the activity label of *standing* is confused with *ironing* and only in three cases in test data.

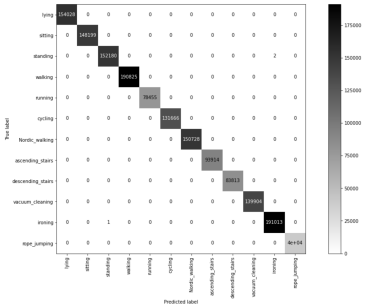


Fig. 14. Confusion matrix of the optimized LSTM

7 Conclusion

The focus of our work is to predict the activity of each subject by using sensor data. We followed the machine learning workflow from the book[13] and processed the large PAMAP2 data set with good results.

We labelled subject's activity with different classification algorithms and conducted comparative experiments, the Optimized LSTM model had the best accuracy. Previous work has been based on models such as decision tree, logistic regression, etc. Our innovation is that we use LSTM to deal with multivariate time series, depending on the complexity of our data set. We have also improved the model by adding the dense layer and dropout, which greatly improves the accuracy of the training and test sets.

For future work, we would use the supervised learning approach to predict people's body mass index (BMI). This will be compared to standard values and used to assess their level of fitness. We hope that through our predictions people can quantify themselves and get a healthier lifestyle.

References

1. Archive.ics.uci.edu. (2012). UCI Machine Learning Repository: PAMAP2 Physical Activity Monitoring Data Set. [online] Available at: http://archive.ics.uci.edu/ml/datasets/pamap2+physical+activity+monitoring [Accessed 23 Jun. 2022].
2. Attila Reiss, Department Augmented Vision, DFKI, Germany. PAMAP2 Data Set Description of Physical Activity Monitoring Data Set. August 2012. https://archive.ics.uci.edu/ml/datasets/pamap2+physical+activity+monitoring
3. Hoogendoorn, M., Funk, B.: Machine learning for the quantified self. On the art of learning from sensory data (2018).
4. Kaggle, Kiran Karthikeyan. Physical_activity_monitoring_using_IMU. https://www.kaggle.com/code/kirankarthikeyan/physical-activity-monitoring-using-imu
5. Barbato, G., Barini, E., Genta, G., Levi, R.: Features and performance of some outlier detection methods. Journal of Applied Statistics 38(10), 2133–2149 (2011)
6. Gao, J., Hu, W., Zhang, Z.M., Zhang, X., Wu, O.: Rkof: Robust kernel-based local outlier detection. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) Advances in Knowledge Discovery and Data Mining. Springer Berlin Heidelberg (2011)
7. A. Reiss and D. Stricker. Introducing a New Benchmarked Dataset for Activity Monitoring. The 16th IEEE International Symposium on Wearable Computers (ISWC), 2012.
8. A. Reiss and D. Stricker. Creating and Benchmarking a New Dataset for Physical Activity Monitoring. The 5th Workshop on Affect and Behaviour Related Assistance (ABRA), 2012.
9. Larsen R T, Wagner V, Korfitsen C B, et al. Effectiveness of physical activity monitors in adults: systematic review and meta-analysis[J]. bmj, 2022, 376.
10. Matthews, Charles E et al. “Best practices for using physical activity monitors in population-based research.” Medicine and science in sports and exercise vol. 44,1 Suppl 1 (2012): S68-76. doi:10.1249/MSS.0b013e3182399e5b
11. Hasan B M S, Abdulazeez A M. A review of principal component analysis algorithm for dimensionality reduction[J]. Journal of Soft Computing and Data Mining, 2021, 2(1): 20-30.
12. DeZyre. (n.d.). Principal Component Analysis Tutorial. [online] Available at: https://www.dezyre.com/data-science-in-python-tutorial/principal-component-analysis-tutorial [Accessed 23 Jun. 2022].
13. Hoogendoorn M, Funk B. Machine learning for the quantified self[J]. On the art of learning from sensory data, 2018.
14. Understanding of LSTM Networks, 25 Jun, 2021 https://www.geeksforgeeks.org/understanding-of-lstm-networks/