Accelerometer Signals Classification Methods for Activity and Still Patient's States Recognition

Abstract

In this project we consider two functions: Activity and Still Patient's State Recognition, that can benefit from Internet of Things (IoT) capabilities in particular for in-home treatments. The designed algorithm is aimed at detecting if a patient is still or patient is having some activity (walking, running, going up/down the stairs or cycling).

The considered methods are:

- Linear classifier with Stochastic Gradient Descent with hinge loss function and L2-regularization;
- XGBoost (eXtreme Gradient Boosting) algorithm.

The Goal

The two algorithms are based on processing and classification of signals with length 4 sec provided by smartphone accelerometer. As the result, there is a detection of patient's states:

- still patient state
- some activity state (walking, running, going up/down the stairs or cycling).

Related work

[1], that was done by the initiators of the experiment, aim to classify the different states of patients with the same accelerometer signal recordings. Features used in the work are: mean, standard deviation, number of peaks for each of Cartesian Axes and Kinetic Indicator. The prediction was made both for activity and movement recognition for multiple classes. The considered methods for prediction are SVM, decision trees (DT), dynamic time warping (DTW) .

Data Structure

Raw accelerometer data with different types of activity is presented in Matlab file format (AR.mat) as a dictionary that contains annotations for each activity recording and its record with approximately 10 minutes length. Each record consists of approximately 66 frames with the length 4 seconds. These short frames are gathered as input data to our algorithms. Each accelerometer point is defined with 3 Cartesian Axes - (x,y,z).

Number of recordings in our data is 319 with 6 types of activity: idle (almost 1% of data), still (standing or sitting) - 5.9%, walking - 7.8%, running - 5.6%, walking down or up the stairs - 74% and cycling - 5.3%.

Overall number of frames is 14 047. The algorithm uses separate labeled frames for training and prediction.

Feature Extraction and Engineering

Features extracted for the algorithms are:

- number of peaks for x,y and z axes, calculated as in [1]
- kinetic indicator - the energy of the accelerometer frame, calculated as in [1]
- mean for x,y and z axes
- median for x,y and z axes
- quantile 0.25 and 0.75 for for x,y and z axes
- standard deviation for x,y and z axes

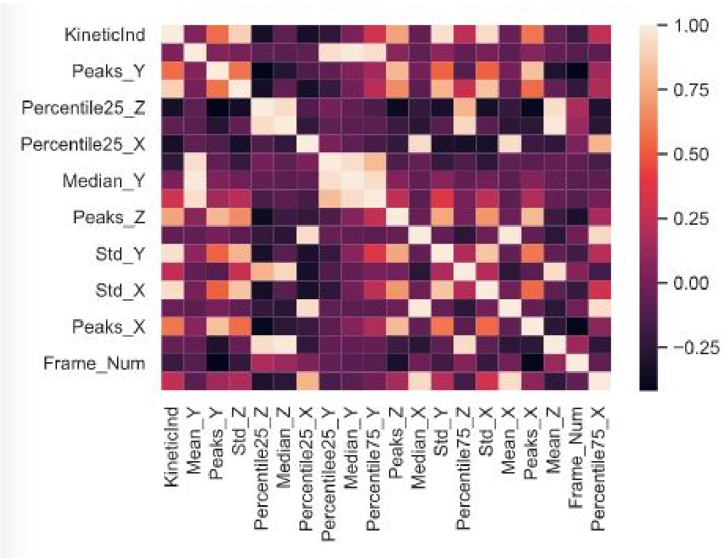
Overall, number of features is 20.

Summary statistics for each feature is presented in Jupyter notebook document in dataframe.

Feature Manipulations:

- Some values of feature Kinetic Indicator was not calculated due to the square root in its calculations that doesn't handle negative values. These values were imputed with the mean value for a column of dataframe with all Kinetic Indicator features for all 14 thousands frames.
- Values with label is patient's state is still or not has been created (binary - 0/1).

Correlation matrix was created to check all connections between features.



It shows that many features of dataset are highly correlated that can affect Machine Learning algorithms (such as linear models). Dimensionality reduction will be applied to overcome this problem.

Data Balance

Data is unbalanced:

- 19 % of data is Still Patient Position (positive class)
- 81% is Activity State (negative class).

Special metrics for tuning, cross-validation and evaluation of models will be applied to overcome class imbalance.

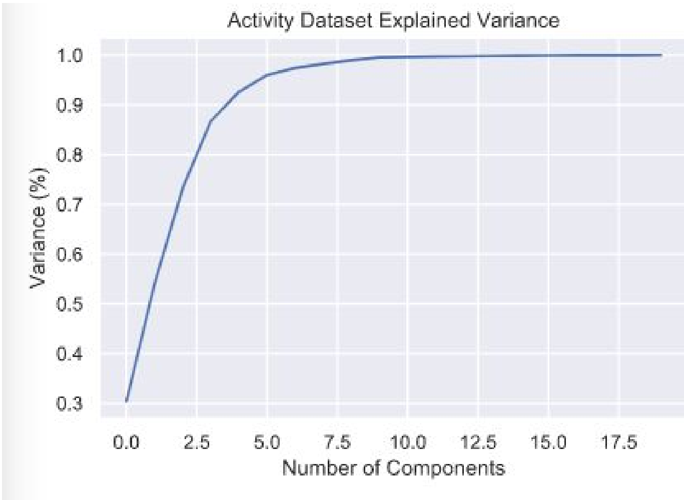
Patient State Models

1. **<u>Linear model</u>**

To avoid multicollinearity and reduce the dimension of features **Principal Component Analysis (PCA)** is used. This approach considers removing correlated features using Single Value Decomposition and is an unsupervised approach.

PCA is affected by scale so feature standardization is applied before applying PCA. scikit-learn module **StandartScaler** was applied to scale features to mean=0 and standard deviation=1.

The best number of components for PCA was chosen using the dependency of the cumulative sum of explained variance ratio to the number of components.



According to the curve the number of components = 9 was selected. This number preserves almost 99% of the total variance of data. The initial number of features = 20 was transformed to 9 principal components.

After removing correlated features, linear classifier model can be applied.

Stochastic Gradient Optimization was chosen due to its efficiency, speed, ease of implementation and possibility of online learning. scikit-learn module SGDClassifier was used for model generating.

Tunning

Parameters were chosen using tuning with GridSearch cross-validation with accent on metric **F1-score** and custom **StratifiedShuffleSplit** to balance ratio of classes in validation set as in the original. F1 metric is the harmonic average of the precision and recall and is good for the imbalanced data in our case.

Model Parameters chosen by tuning:

- Regularization constant alpha - 0.001,
- Loss function - Hinge loss - (soft-margin) linear function for Support Vector Machine ,
- L2 regularization,
- Number of iterations - 9.

Accuracy metrics

For accelary assessing such metrics and methods were used:

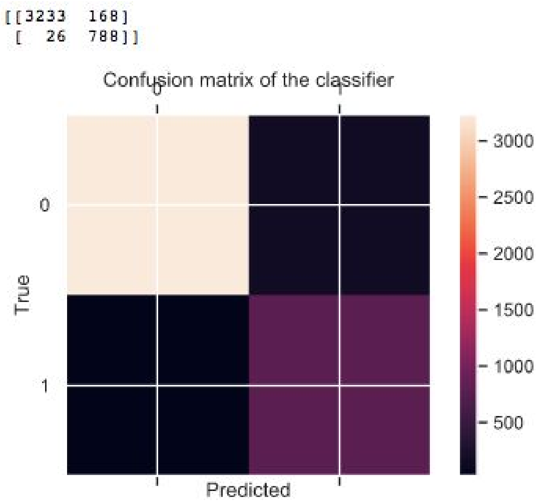
- building confusion matrix to assess typical errors for algorithm
- F1-score
- Precision and Recall
- The ROC-AUC metric
- Matthews Correlation Coefficient (extremely good metric for the imbalanced classification)

Model Evaluation

Accuracy metrics of algorithm was calculated using holdout test set with 30% of data

Metric	Value
F1-score	0.89
Precision	0.82
Recall	0.97
Roc-Auc	0.96
Matthews Correlation Coefficient	0.87

Confusion matrix:



- The **confusion matrix** shows that high part of errors are FP (false positive), thus precision is not good enough (0.82). On the other hand, recall is pretty high (0.97) that means that we predict almost all our objects from positive class (Still State) that is very good for our imbalanced dataset. Harmonic average F1-score is 0.89. This metric was used as scoring metric during model tuning.
- **Roc-Auc** is high (0.96) that means that the classifier has high prediction ability.
- **Matthews Correlation Coefficient** is good metric for the imbalanced classification and can be safely used even classes are very different in sizes as in our case. It involves values of all the four quardants of a confusion matrix, it is considered as a balanced measure. It ranges between −1 and 1. The metric in our model is 0.87.

At the end, whole **pipeline** was constructed for combining:

- feature standardization
- PCA
- model training and predicting

2. Gradient Boosting (GB)

In case of GB, it won't be caught wrong when dealing with very highly correlated input features. Due to its nature, if the number of trees is set to an imaginary infinite value, for two perfectly correlated variables, GB will pickup exactly 50%|50% each features. **XGBoost** model was chosen for for classification of Still State of patient.

The default base learners of XGBoost are tree ensembles. The tree ensemble model is a set of classification and regression trees (CART).

The best model was picked using **GridSearch 5-fold cross-validation** again with metric **F1-score** to maintain consistency.

The model parameters according to tuning:

- maximum depth of each base tree algorithm is 3
- minimum child weight (sample size threshold in the node for stop splitting) - 1

Model Evaluation

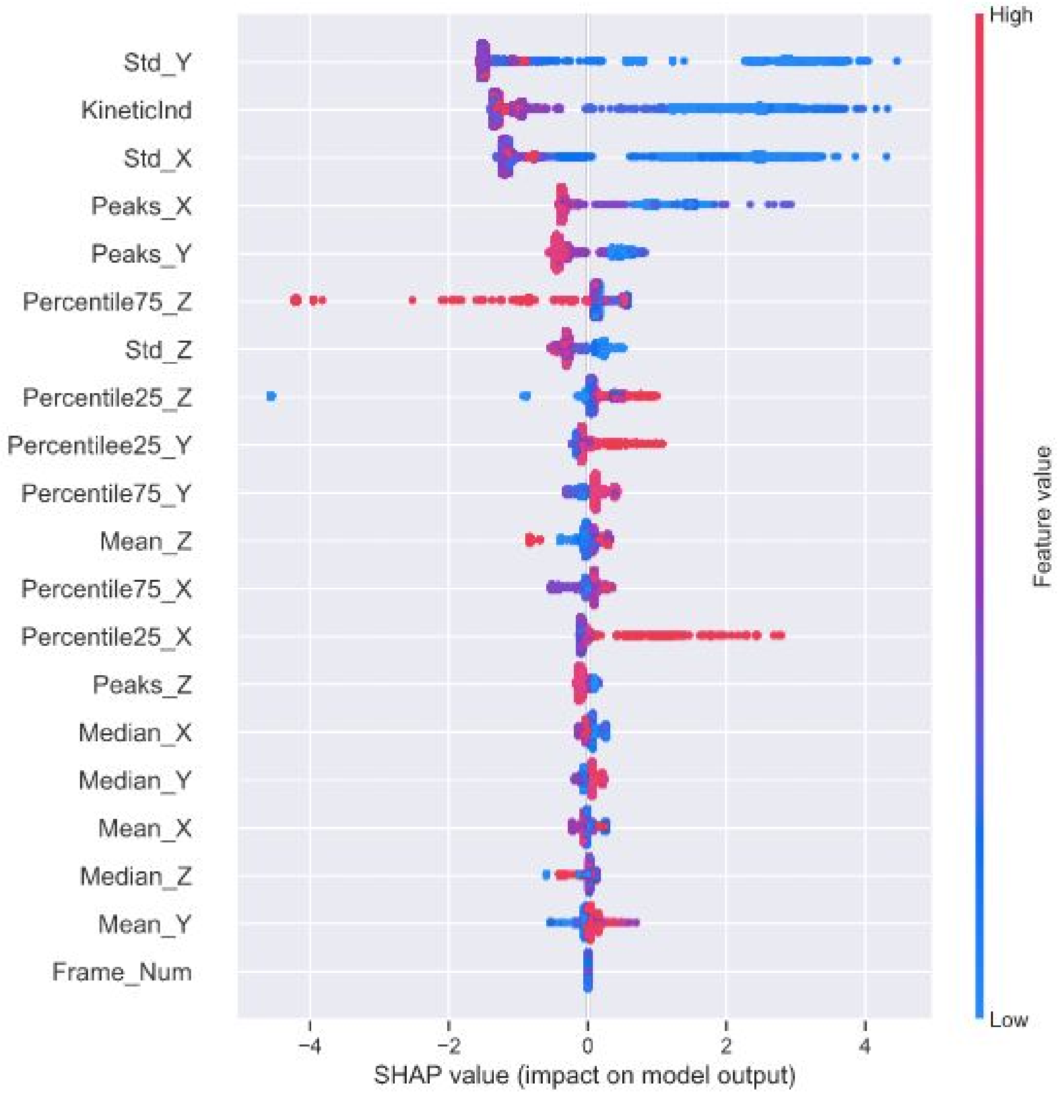
Accuracy metrics of algorithm was calculated using holdout test set with 30% of data/

Metric	Value
F1-score	0.996
Precision	0.998
Recall	0.995
Roc-Auc	0.997
Matthews Correlation Coefficient	0.995

All metrics are very high that means that prediction ability of the model is very strong.

Feature importances using XGBoost

The built-in feature importance orderings are very different for each of the options provided by XGBoost (gain, cover and weight). That is why **Tree SHAP method** was used to assess feature importance for every measurement in the dataset.



Every activity measurement has one dot on each row. The x position of the dot is the impact of that feature on the model's prediction for the measurement, and the color of the dot represents the value of that feature for the measurement.

The features are sorted by mean(|Tree SHAP|) and so we see the **standard deviation for y-axis, Kinetic Indicator and Standard deviation for x-axis** are the strongest predictors of labeling signal as recorded during still position (sitting/standing).

By plotting the impact of a feature on every sample there are important outlier effects. For example, while **0.7 percentile for z-axis** and **0.25 percentile for x-axis** are not the most important features globally, it is by far one the most important features for a subset of signals (red color - high feature value).

Conclusions

The project contains two Patient's State Prediction Models that are trained to classify is patient's state is still or active.

Linear classifier with Stochastic Gradient Descent with hinge loss function and L2-regularization uses Principal Components analysis to reduce data multicollinearity and shows moderate accuracy metrics: F1-score is 0.89. At the same time, this algorithm correctly extract almost all samples from positive class (recall=0.97) but recognize some negative samples as positive (precision=0.82). Matthews Correlation Coefficient that handles all quadrants of confusion matrix is 0.87. That is average result. But algorithm has a high prediction ability according to ROC-AUC score (0.96).

As an alternative to linear methods, eXtreme Gradient Boosting algorithm was used as a very efficient and high-speed model to deal with multicollinear features. Accuracy metrics achieved are very high that means that model correctly deal with the task of Patient State Classification.

Using XGBoost and SHAP method feature importance map was constructed to assess feature importance for every measurement in the dataset. The most important features according to this method are: standard deviation between values of x and y-axis and kinetic indicator of signal. Also, regarding some subset of signals features with 0.25 percentile of x-axis values and 0.75 percentile of z-axis values are also important.

Suggestions to future work:

- using more features in parameters grid for tuning both algorithms
- trying upsampling or downsampling to balance class ratio and improve classification ability of both algorithms
- use dataset with different movement signals recordings to improve algorithms and train it to recognize still position also from different movements of patient (such as simple hand and leg movements)
- try other optimization options in linear algorithm
- create classifiers for other patient states
- extract more features from signals - such as entropy, skewness, kurtosis, signal magnitude
- add frequency domain analysis such as Fourier transform and extract more features from signals and metrics for AC and DC components of signal

References:

1. The initial research is described in the paper: [Enabling IoT for In-Home Rehabilitation: Accelerometer Signals Classification Methods for Activity and Movement Recognition]