Predicting Daily Activities with Smartphone Sensor Data

Rowana Ahmed, Ben Shea, Mukund Poddar, Saul Holding, Nellie Ponarul

Introduction

The ability to accurately track and identify a person's daily activities is highly valuable for many areas of medical research, like clinical trials, disease progression, and surgery recovery. While there are several wearable devices on the market to track this kind of data, smartphones are becoming the most commonly carried inertial sensor-enabled devices.[1] In fact, 85% of the US population owns a smartphone.[2] This makes passive sensor data much easier to collect, helping create unique biomedical research opportunities. For example, smartphone data has been used to quantify the mobility and social interaction of patients after cancer surgery and mobility after a spinal cord injury.[3]

We want to determine how effectively smartphone sensor data can be used to record and identify daily activities like walking, sitting, and standing. We will compare a number of prediction algorithms and determine the best model to predict these activities. From monitoring one's own physical activity level to the detection of the onset of a seizure episode for epileptic patients, the insights gained from this analysis could be used to motivate the use of the ubiquitous smartphone in human activity recognition in a plethora of fields.

Methods

Movement data collected during daily activities is publicly available from the UCI Machine Learning Repository.[4] This dataset was collected from 30 volunteers performing six daily activities (standing, sitting, laying, walking on a flat surface, walking upstairs, and walking downstairs) for approximately 15 minutes.[5] The sensor signal data was acquired from a waist-mounted Galaxy S II smartphone worn by the subjects during these activities. Input from the accelerometer and gyroscope produced data at 50Hz, which was sampled in windows of 2.56sec with 50% overlap.

---

[1] Patel et al, A review of wearable sensors and systems with application in rehabilitation. Journal of NeuroEngineering and Rehabilitation, April 20, 2012

[2] Pew Research Center Mobile Fact Sheet, April 7, 2021

[3] Mercier et al, Digital Phenotyping to Quantify Psychosocial Well-Being... : American Journal of Physical Medicine & Rehabilitation, December 2020

[4] UCI Machine Learning Repository: Human Activity Recognition Using Smartphones Data Set

[5] Anguita et al., Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine. Ambient Assisted Living and Home Care 2012.

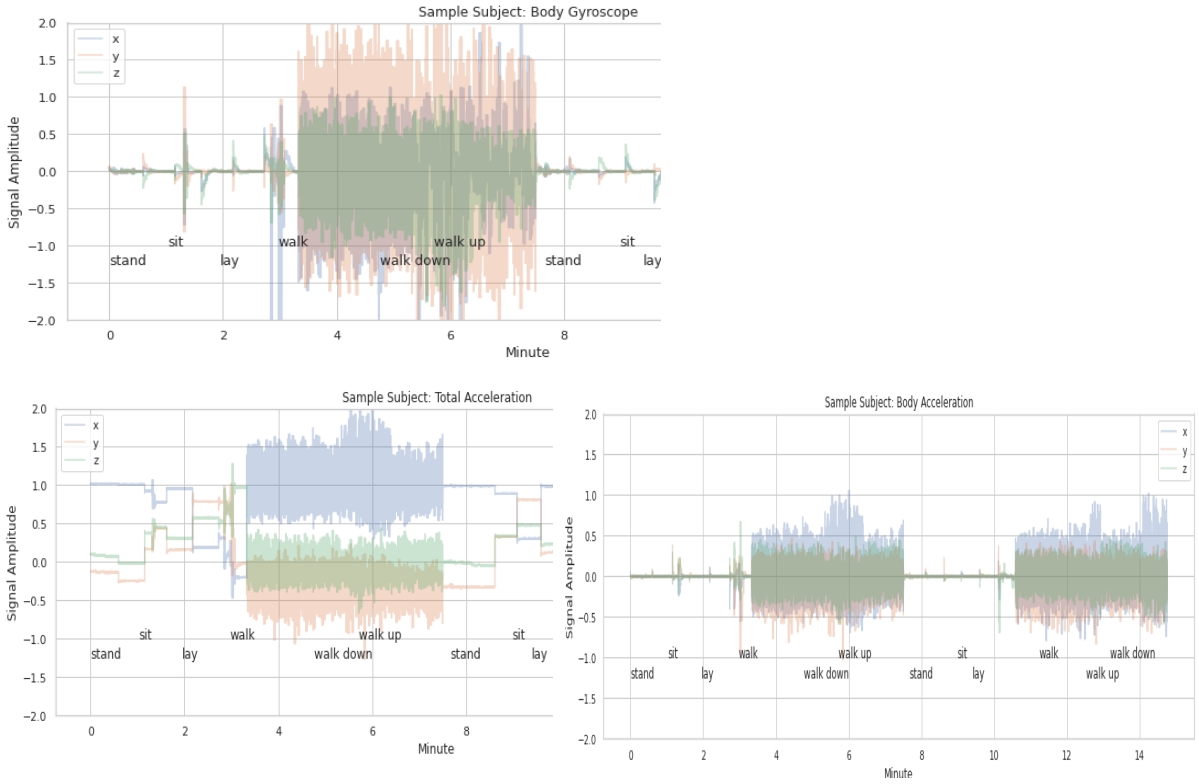

Figure 1: Accelerometer and Gyroscope signals for Sample Subject

The figures above visualize the raw signals from the accelerometer and gyroscope sensors in the x, y, and z planes for a sample subject. The dynamic movements (walking on a flat surface, walking upstairs, and walking downstairs) have similar signal patterns, and the stationary movements (sit, stand, lay) also share similar signal characteristics. The ideal model would be able to find nuanced differences within these two categories in order to distinguish among the specific dynamic and stationary activities.

Each data sample corresponds to a 2.56 second window during which a subject is performing one of the six activities. Across the 30 subjects, there are 7,352 training data points and 2,947 test data points available for modeling. 17 accelerometer and gyroscope signals were calculated from each window using the time and frequency domains (Fast Fourier Transforms were employed to process the frequency data). Each signal was then aggregated by the UCI researchers using various statistical measures (e.g. mean, standard, deviation, max) to derive 561 hand-engineered features that serve as predictors for our models. The feature matrix was normalized to range in values from -1 to 1.

We assessed the classification accuracy using both parametric (LDA) and nonparametric approaches (KNN, SVC, and tree based models). Generally, we use cross-validation and grid search for all methods to find the optimal hyperparameter combination that maximizes classification accuracy. In addition to optimizing for accuracy, we also consider which method is the most computationally efficient so that it could be practically deployed and classify motion in real-time on a smartphone or wearable device.

Results

Principal Component Analysis
Given the large number of predictors we have in the model, we performed a principal component analysis to understand how our data relates to the output classifications.

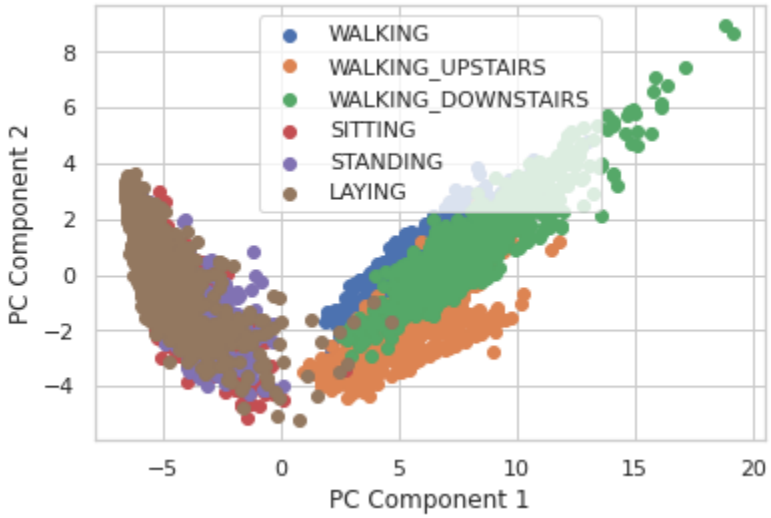

Figure 2: Principal Component Analysis of all Predictors

The visualization of the first two principal components in Figure 2 above reveals clear separations between actions that are stationary (sitting, standing, and walking), and those that are active (walking, walking upstairs, and walking downstairs).

Models

Linear Discriminant Analysis (LDA)
The first model we tried is Linear Discriminant Analysis, which is a parametric model assuming that the predictors follow a Gaussian distribution. We used the implementation of LDA from the Python package sci-kit learn. LDA requires no hyperparameters so no tuning was required. We tried applying PCA (with varying numbers of principal components) on the training set but it did not improve results. Since LDA can be used for dimensionality reduction, this result was not too surprising. The scaled data could be used directly for accurate feature importance analysis. We found that tBodyAccjerk-energy()-Z, tBodyAccjerk-energy()-X, and tBodyAccjerk-energy()-Y were the 3 most important features in determining the predictions from the LDA model. More generally, body acceleration and make up the most important features.

The f1-score from the LDA model was 0.963 and the test accuracy was 96.2%, indicating that the data is indeed multivariate normally distributed. Most of the classification errors were made

between sitting and standing, which is likely because both of these are static positions with similar signal patterns. The other set of errors were between the three walking activities ('Walking', 'Walking Upstairs', and 'Walking Downstairs').

Support Vector Classification (SVC)

We also considered more flexible, non-parametric models to see if performance is improved. We implemented the support vector classification (SVC) model using the Python package sklearn. To fit the model, we performed 10-fold cross validation to tune the following hyperparameters: squared L2 regularization parameter, C, and the kernel coefficient gamma. C determines the number and severity of the violations to the margins that we will tolerate. The results of the hyperparameter tuning gave us a squared L2 regularization parameter of 1000 and a kernel coefficient of 0.0001. We used these parameters to fit our final model, and achieved a test set accuracy of 96.2% and f1-score of 0.962.

Similar to the LDA model, the confusion matrix shows that the misclassifications usually happened among the activities "Walking," "Walking Upstairs," and "Walking Downstairs," as well as between "Standing" and "Sitting." We evaluated the ROC based on a binary condition (e.g. classifying "Standing" correctly or misclassifying with any of the other activities) and saw that across all activities the Area Under the Curve (AUC) was very high.

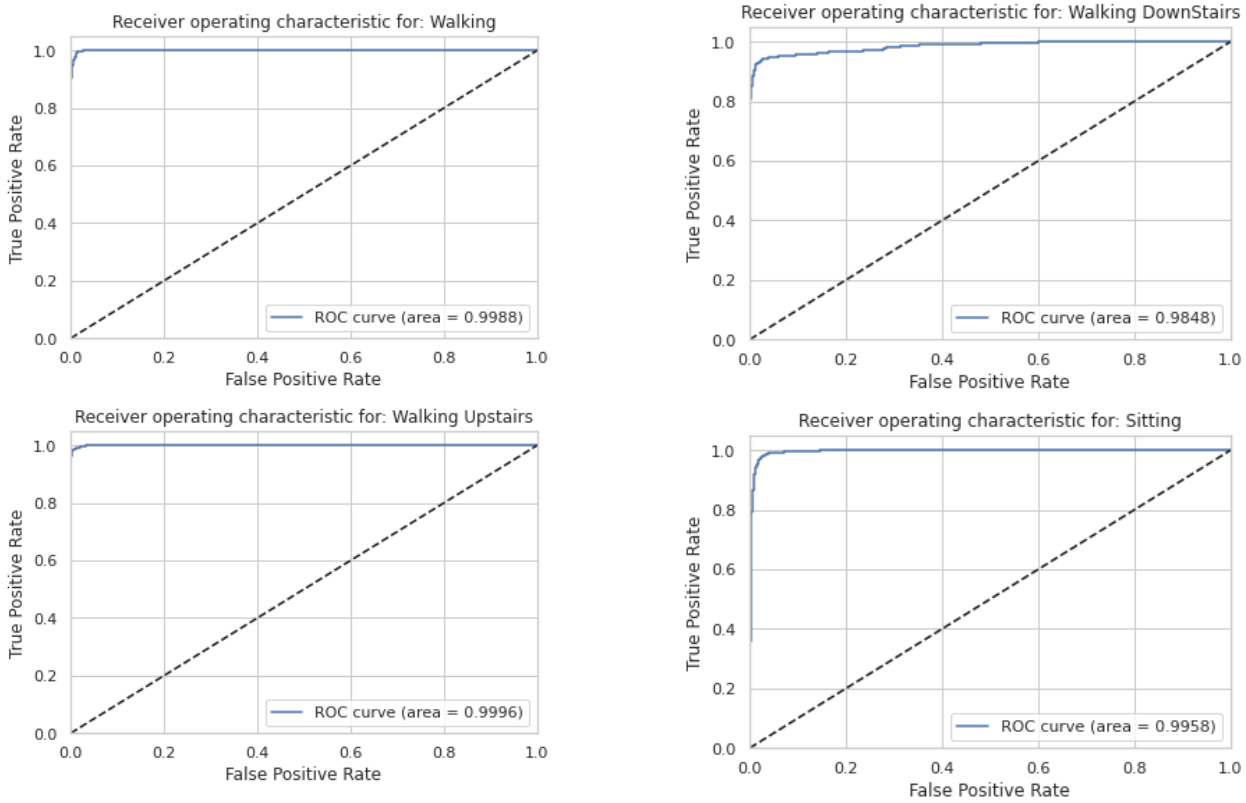

Figure 3: ROC curves for SVC model

K-Nearest Neighbors Classifier (KNN)

A KNN model was fit to the training data using the Python sklearn package. Because the feature matrix has been normalized, the predictors were without any additional processing. 10-fold cross validation was used to find the optimal k value of 8 neighbors, which resulted in a final model that had a 90.7% classification accuracy on the test data set.

Because KNN is known to perform poorly in high dimensional space, the model was also fit in a reduced eigenspace using PCA. We looked at principal component scores ranging from the first 2 PCs up to the first 100 PCs (accounting for ~95% of total data variance). However, after using 10-fold cross-validation to determine the optimal k in each of these reduced feature spaces, the test accuracies were lower than what was achieved in the original feature space.

The errors made by the KNN classifier generally occur when 'Sitting' is being misclassified as 'Standing,' although interestingly, the reverse error occurs less frequently (i.e. 'Standing' is not misclassified as 'Sitting' as often). This may indicate that there is a broader distribution of feature values associated with 'Sitting' than with 'Standing,' which seems reasonable as posture can vary widely while individuals sit. Additionally, there are some misclassifications among the dynamic movements.

Random Forest

Again using Python's sklearn package, the first ensemble tree model we implemented was a random forest model. To tune the different hyperparameters such as maximum depth, maximum number of features used at each split, the number of trees, and whether or not to bootstrap, we performed a 3-fold randomized cross validation. The tuning indicated that we should bootstrap with the maximum depth as 81, the maximum number of features as 9, and the number of trees as 540. These parameters were used to fit the final model on the training set.

Evaluating the model, the random forest gave a test accuracy of about 94% and an f1-score of about 0.937. The three most important features are tGravityAcc-min()-X, angle(X,gravityMean), and tGravityAcc-mean()-Y. More generally, gravity acceleration and angle make up the most important features. Ultimately, the model performed well, but did not outperform our baseline LDA model. The model had the most difficulty differentiating between walking up and down stairs and differentiating between standing and sitting.

Gradient Boosted Tree Model (XGBoost)

The second ensemble tree method we implemented was a gradient boosted tree model, using the Python package XGBoost. To fit a gradient boosted model, we performed 3-fold cross validation to tune the parameters learning rate, L1 regularization parameter, and the maximum tree depth. The results of the hyperparameter tuning gave us a learning rate of 0.429 and a L1 regularization parameter of 0.778, and a maximum tree depth of 90. We used these parameters to fit our final model. Similar to other models, most of the misclassification errors were between the three walking categories and between sitting and standing.

The gradient boosted model trained with our final hyperparameters gave us a test set accuracy of 91.9%. This is far lower than the test set accuracy of the LDA , SVC, and random forest

models. This is likely due to the fact that the gradient boosted model learns in an additive manner by adding a decision tree one at a time to minimize the residual sum of squares, which makes it learn a complex classification boundary. The high performance of LDA provides strong evidence that the true classification boundary is linear and is not complex, thus the gradient boosted model has more variance in the model than it should, leading to unstable predictions and poorer test performance. Although the random forest model is also learning a complex boundary, it is averaging the predictions of an ensemble of trees which means there is likely less variance in the predictions it is making, making it a superior ensemble tree model for this data.

The three most important features in the gradient boosted tree model were tGravityAcc-mean()-X, tBodyAcc-arCoeff()-Z,4, and tGravityAcc-min()-X. More generally, gravity acceleration and body acceleration make up the most important features. We determined most important features by the number of splits in which each feature was used across all trees used in training the model.

Table 1. Comparison of Predictive Models Test Performance

Model	Test Set Accuracy	F1 Score
LDA	0.962	0.963
SVC	0.962	0.962
KNN	0.907	0.905
Random Forest	0.939	0.937
XGBoost	0.919	0.917

Discussion

Among the tested models, Linear Discriminant Analysis had the best performance, with an f1-score of 0.963, followed closely by the Support Vector Classifier with a score of 0.962. KNN had the lowest performance (at a still respectable 0.9), and the Random Forest and XGBoost tree based models fared slightly better with f1-scores of 0.937 and 0.917, respectively. LDA's superior performance may be attributed to the model's inherent assumption that the predictors are multivariate Gaussian distributed, which is a reasonable modeling assumption as most people have similar movement styles within a range of normally distributed values. The PCA plots above also indicate a multivariate Gaussian pattern. Thus, the LDA model might have the lowest bias among all classifiers tested.

The fact that both the Support Vector Classifiers and LDA performed well, while the tree-based models and KNN did not, also suggests the boundaries in the feature space are almost linear with clear separation in the clusters. Hence the additional flexibility of the nonlinear models might not be helpful in our problem, and may be adding high variance in our results due to

additional and unnecessary flexibility. The computational efficiency of the LDA versus these non-parametric models further validates the benefits of using a parsimonious model to identify activities quickly in real-time.

These findings, while only assessing a limited number of activities, can further expand the usage of phones and wearables in monitoring and analyzing human activity and the effects these activities can have on one's health. These findings can be broadened and potentially used in robotics for human behavior and surveillance, among other things.

Limitations

The data collection method used is a limitation in our study and might reduce the high prediction accuracy achieved by these models. The subjects in this study wore a smartphone device mounted to their waist in a very controlled manner, which does not generalize to how the broader population carries their smart device. For example, due to women's clothing design, many people do not have waist level pockets that can accommodate a smartphone. When phones are carried in purses or elsewhere, the accelerometer and gyroscope signals may vary substantially, resulting in very different features that could impact classification accuracy. Thus, more controlled wearables, such as smartwatches, may be easier to calibrate across users and are preferable for activity classification tasks.

Ethical Considerations

Our group strongly believes that an assessment of algorithmic fairness stemming from the used data, the implemented model, and the final predictions should be the norm in papers such as these. As such, we attempt to present an honest interpretation of the algorithmic fairness of our work.

We believe that there are at least two biases that may arise from our model. Given the nature of our work, both of these biases stem from the data itself. First, as there are no individuals with disabilities in our dataset, our model would be biased against individuals suffering from physical disabilities if it is used to predict human activity for this demographic. For example, our model would likely inaccurately predict the type of movement of someone using crutches. Moreover, as the data was gathered with waist-mounted smartphones, there may be a bias in favor of individuals who are more likely to carry their phones in this manner (e.g. older individuals or males). Again, further analysis and scrutiny on the creation of the dataset should be done to help identify the actual biases and to ultimately mitigate them.

In the end, these potential biases warrant further investigation as they can lead to unintentional but dire consequences, especially if intersectional biases exist such as a bias against people who are young or physically disabled. Hence, it is this paper's recommendation that, as a next step, a rigorous investigation into how algorithmically fair our model is needs to be conducted. If issues are found, we recommend that a more diverse dataset be used to generate a fairer model. Indeed, the ability to predict movement for different demographics, especially those with physical disabilities, provides the opportunity not only to offer an equitable movement monitoring software but, ultimately, to save lives.

Future Work

In future work we would like to see if the accuracy we achieved with our predictive models could be achieved with less predictors. Having less predictors would cut down on the computational cost of this model and make it less memory and time intensive to generate and store the predictors necessary to run our models. In addition to making our models more parsimonious, reducing the number of predictors would reduce the variability in the predictions and make them more stable. We would also like to test this method out on more activities, like running or biking in order to determine how robustly the models can predict daily human activities from smartphone sensors. Being able to accurately predict daily activities using these low-cost, ubiquitous sensors can benefit patients in a variety of contexts, including detecting falls, managing post-operative rehabilitation, and encouraging active and healthy lifestyles.

Member's Contributions

For the project outline, Nellie wrote the background, Mukund wrote the description of the dataset, Saul wrote the scientific objectives, and Rowana and Ben wrote the methods considered for analysis.

All members contributed to writing the final paper, creating the slide deck, and presenting the results. The modeling was conducted as follows:

- LDA - Mukund Poddar
- SVC - Ben Shea
- KNN - Rowana Ahmed
- Random Forest - Saul Holding
- XGBoost - Nellie Ponarul