Human Activity Classification via Smartphone

University of California, Santa Barbara

Kevin Ayala , Megan Handa, Nathan Hwangbo, Dennis Wang

Table of Contents

Executive Summary

The purpose of this report aims to analyze the predictive power of smartphones in human activity recognition based on data gathered from thirty individuals who performed six different activities with a Samsung Galaxy smartphone strapped to the waist. Our project seeks to develop a model which can accurately classify the activity that a user is performing with consideration to the duration of run time and ease of interpretability. Additionally, we seek to identify feature variables with the greatest predictive power in classification. First, the datasets were merged and preprocessed for data cleaning. This was done to ensure proper datasets were obtained that did not have missing entries or incorrect alignment due to human errors and missing assumptions. This is followed up with exploratory data analysis to gain insight about our data including dimensionality reduction and class imbalance visualization. Through data processing and exploratory analysis, ultimately three models were build that addressed the predictive power based on the dataset. Through hyperparameter tuning, ultimately a multinomial logistic regression was selected for its optimal performance and ease of interpretability of the outcomes. The predicted values provide a promise in sharpening the power of machine learning, industry development and assessment, as well as improvement research in statistical methods.

Introduction

As devices like the Apple Watch, Fitbit, and other smartwatches grow in popularity, it is increasingly important that our machines accurately classify the activity that a user is performing. This activity classification has obvious applications in fitness, but the same models could also contribute to the healthcare and surveillance fields. Our project seeks to develop a model which can accurately classify the activity that a user is performing with consideration to the duration of run time and ease of interpretability.

For the purpose of this report, we will define human activity recognition as the ability to distinguish between the following static and dynamic states of physical activity: standing, sitting, lying, walking on flat ground, walking upstairs, and walking downstairs. We developed our models using data from a pre-existing dataset from the University of California, Irvine repository which collected gyroscope and accelerometer readings from thirty participants with Samsung Galaxy phones strapped to their waist as they performed the six states of activity.

Data Description

Our dataset consists of 10929 observations with 561 variables, collected from thirty volunteers from ages nineteen to forty eight and manually labeled. Each subject performed three static activities (standing, sitting, and lying) and three dynamic activities (walking, walking downstairs and walking upstairs). The experiment also collected data during postural transitions that occurred between the static postures(stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie, and lie-to-stand). This activity classification variable served as the response variable in our model, with the remaining 560 variables used as predictors. This data was collected from a

Samsung Galaxy S II strapped to the waist which recorded three-axial linear acceleration and three-axial angular velocity at a constant rate of fifty hertz (Anguita Et al 1).

The raw data was not publicly available - noise filters were applied to all measurements and sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). Additionally, a Butterworth low-pass filter was used to separate the sensor acceleration signal into body acceleration and gravity. There was also an assumption that the gravitational force would have only low frequency components, therefore, a filter with 0.3 Hz cutoff frequency was used (Anguita 1 Et al).

Data Preprocessing

Though the data files we obtained from the University of California, Irvine machine learning repository were not "raw", we faced a number of challenges that made preprocessing one of our most time consuming steps. Our main tasks stemmed from combining files, checking for missing values, treating outliers, scaling data, and removing repetitive variables.

The dataset was originally comprised of twenty seven files, separating important identifiers such as headers, subject identification numbers, activity labels, training data, testing data, inertial signals, acceleration measurements, and more. We approached this problem by identifying which observations in individual files referred to the same subject and activity, then used SQL statements to merge the files to create a master dataset. Next, we completed checks to ensure that our dataset had no missing entries. Though we found that we had a complete dataset, this is an important step in data preprocessing to determine if imputation methods must be considered.

After checking for completeness of the data, we began an analysis of outliers. We initially used a standard approach of identifying an outlier as a point with a value outside one and a half times the interquartile range, which left us with large amount of outlier points. Given that we began with 10929 observations, we felt that this was too many observations to denote as outliers. We experimented with changing the bounds to 1.75 times the interquartile range, which also resulted in too many outliers to justify exclusion. We sorted the outliers by subject identification number to see if we could identify one subject whose measurements were being marked as "outliers" due to differences such as a different walking style or postural angle, but we were unable to draw any significant conclusions. We ultimately decided not to remove any outliers from the data, given that the experiment was only performed with thirty subjects but our goal of activity classification is meant to generalize to a smartphone-owning population of two billion people ("Number of Smartphone Users Worldwide 2014-2020."). In future studies, it would be interesting to explore how we define an outlier in the context of activity recognition, but due to time constraints, we proceeded to move on to scaling.

For the purpose of our model, we decided to scale the variance, and not the mean, since our features were not all measured on the same scale. Our original dataset contained over five hundred variables, indicating that dimensionality reduction would be a necessary step for our model's ease of interpretability and fast run time. Principal component analysis (PCA) was our

chosen method of dimensionality reduction, and because PCA is highly sensitive to unscaled features, the step of scaling the training features was crucial.

Finally, upon examination of the observations, we decided to remove the 126 columns containing the phrase ‘bandsEnergy’. The documentation told us that these columns are meant to represent “intensity” in the frequency domain, but there was no information given about how this was measured/calculated. Additionally, many of these columns had the exact same name, with no discernable way to distinguish between them. Based on the documentation, it seemed that this information was already captured in the time acceleration and gyroscope variables, so we decided to remove these columns from the dataset to help avoid multicollinearity (and hence variable importance).

Once we completed combining the twenty seven files, checking for missing data entries, completing outlier detection, scaling feature variables, and removing repetitive variables, our data had 435 features and 10929 observations. From here, we proceeded to split our data into training and testing sets.

Data Splitting

During the preprocessing step of combining data files, we found that the experiment had already designated the data into training and testing sets, with seventy percent of the data denoted for training and the remaining thirty percent for testing. Though it would have been easier to use the given split, we decided that the relatively small sample size of ten thousand observations from only thirty participants justified a split with a larger training set, so that our results could be generalized to the greater smartphone-owning population with more accuracy. Though our model is more likely to overfit than one trained on the seventy-thirty split, we thought this cost is worth the tradeoff to ensure that our model is trained on a sufficiently large dataset. At the end of this step, our training set had 8743 observations, and the test set had 2816.

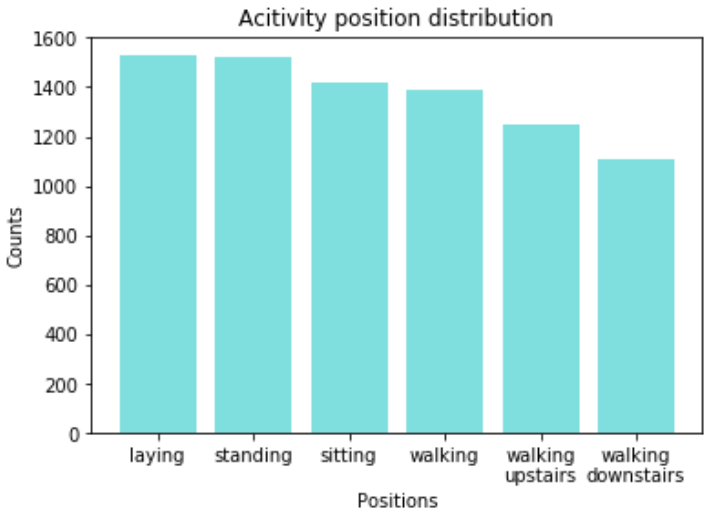
Exploratory Data Analysis

Before attempting to build a model, we performed a number of exploratory data steps to better understand the 435 variables we were working with. We began by grouping the predictors into their respective sensor measurement categories - an incomplete summary of the grouping visualization is shown below.

	count
fBodyAcc	79
fBodyGyro	79
fBodyAccJerk	79
tGravityAcc	40
tBodyAcc	40
tBodyGyroJerk	40
tBodyGyro	40
tBodyAccJerk	40

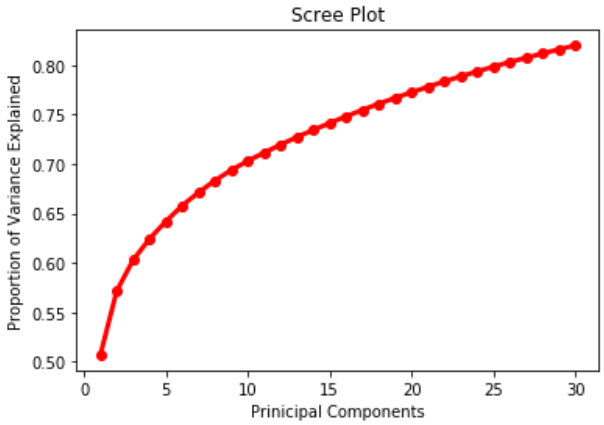
In this table, the left side refers to the measurement category while the right side denotes the number of variables which fall into a subcategory of that measurement. Doing this analysis allowed us to see that the frequency domain variables (categories starting with an f) had more columns in each of the categories, but had fewer categories compared to those measured by time. After separating the data, we moved to data visualization.

We approached our research question with a heavy emphasis on interpretability, and data visualization provided the perfect opportunity to develop meaningful graphs that would communicate the key points of our dataset, even to a non-technical audience. One key point we wanted to address when building a multiclass classifier was the class balance in our data. Because our data was collected at regular time/frequency intervals, we hypothesized that we would have fewer measurements for activities which could be completed faster. For example, the average human walks downstairs more quickly than walking upstairs or on flat ground, which could potentially train our model to predict walking upstairs more than walking downstairs simply because there were more observations of the former. A potential class imbalance could negatively affect predictive accuracy for the less frequent classes, requiring alternative sampling techniques and alternative performance measures of a model.



Upon examination, we found our classes to be almost evenly distributed which eliminated the need for alternative sampling techniques. As we had hypothesized, there were fewer measurements for “faster” activities such as walking downstairs but the difference in the number of observations was too small to significantly impact our model’s predictions.

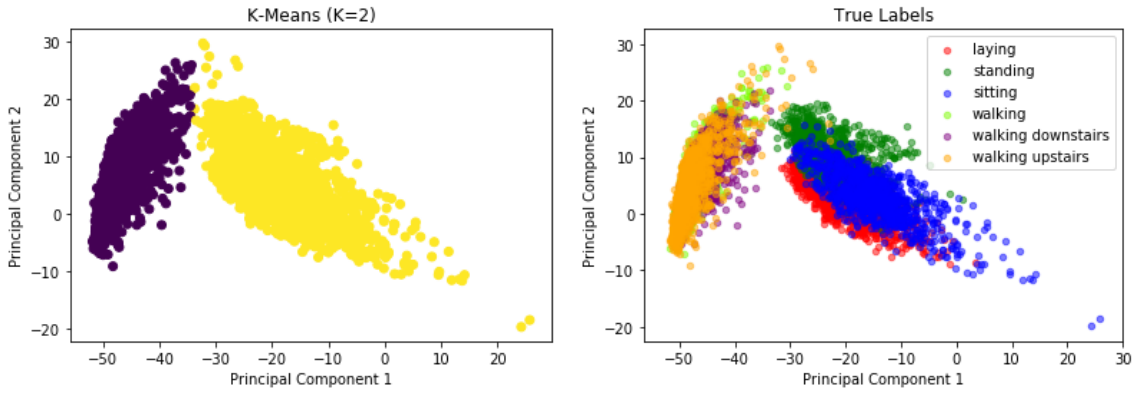
Along with class balance, we felt that it was important to address the challenge of dimensionality reduction in a visualization. Because it is difficult to visualize all 435 features of our dataset, we chose to perform PCA to describe our data, which constructs components to account for the most variance in the data. To decide on the number of principal components required to capture a substantive proportion of the variance, we created a scree plot.



Our scree plot revealed a slight elbow around ten principal components, at which point we account for roughly seventy percent of the variance in the data. We proceeded to model using these ten principal components. However, the scree plot did not reveal a sharp elbow, so for future studies, we hope to do a more detailed analysis in methods to determine the ideal number principal components to use, such as Horn's Parallel Analysis (Glorfeld).

For the purpose of visualization, we used the first two principal components to plot our data. To begin, we conducted a silhouette analysis on a K-means algorithm for k $\in$ [2,10]. This algorithm measures the data's average distance to its own centroid, as well as its distance to the closest neighboring cluster. The silhouette score ranges from [-1,1], where 1 is the best possible score, and -1 is the worst. We plot the results in the scree plot below.

Based on this analysis, it appears that having two clusters by far yields the best fit. To ensure this is a good representation, we compared it against the true labels of the clustering, which is the plot below on the right. We notice that with two principal components, the algorithm picks up the differences between the static and dynamic positions, since the three static labels (laying, standing, and sitting) are grouped together, and the three dynamic labels (walking, walking downstairs, and walking upstairs) are also grouped together.



Models and Evaluations

Throughout the process of building and testing models, we favored classifiers with accurate classifications and efficient run times. We began with a random forest model as a baseline, then tested a one vs. rest logistic regression classifier and then finally a multinomial logistic regression classifier to find that our champion model was a tuned multinomial logistic regression classifier.

Random Forest

To get an idea of variable significance, we binned the features into six categories (based on the category table we generated at the beginning of the EDA section of this report). Then we split our data into six data frames (one for each category), and ran a random forest with 100 trees on each of them. Using this method, we were able to avoid multicollinearity to get a clear picture of variable importance. The table below shows the misclassification error for each category. We see that Acceleration due to gravity (tGravityAcc) and Body acceleration (tBodyAcc) were the two most powerful predictors. We also noticed that the time-domain categories consistently outperformed the frequency-domain categories.

Misclassification error for each category

fBodyACC	0.23959
tBodyGyro	0.25043
tBodyAcc	0.19855
fBodyBody	0.37180
tGravityAcc	0.09752
angle	0.29279

We began our model building through the construction of a random forest model, fitting 1000 trees on the full training set, using the ML library's default parameters. We felt that a random forest classifier would be a good choice for our baseline model because it is a robust non-parametric model. Given ten thousand observations and 435 variables, we wanted to make as few assumptions about the data as possible. To counteract highly correlated variables, a random forest classifier subsets predictors at each split which decorrelates the decision trees used to create the random forest model. Additionally, random forest classifiers do not require the linearity assumptions that other models such as logistic regression requires. This made random forest a safe starting point.

Our random forest classifier performed very well for a baseline model, with an error on the training set of .07 and on the test set of .08. One of the cons of a random forest model is that it is computationally expensive. We attempted to remedy by fitting the model with the ten principal components that accounted for the most variance in the dataset. Our PCA random forest model performed approximately ten percent worse, with a .17 error on the training set and .18 on the test set. We hypothesize that this decrease in accuracy has two causes. First, ten principal components only accounts for roughly 70% of the variance in our original data. Second, principal components point in the direction of the greatest variance, rather than the direction of greatest predictive power, making it less useful for our supervised machine learning problem. However, we thought it was important to note that the runtime using only ten principal components was significantly faster than using every variable, and depending on the application, the loss in predictive power may be a necessary trade off for faster run time.

Additionally, we used the random forest model run on the full dataset to identify which variables were most important in classifying the subject's activities. As we showed earlier, our data is separated between the static and dynamic activity groups - our k-means clustered the two groups using only two principal components. From that point, the best predictors are the ones that can distinguish between activities within the static or dynamic activity groups. Upon examination, our random forest model identified gravity and the angle of movement as the two variables that most determined the state of the user. This conclusion makes sense because when looking at the three dynamic activities (walking, walking upstairs, and walking downstairs), the main difference between them is the direction of motion. which would result in varying values of acceleration due to gravity, and direction of movement. For ease of interpretability in our variable importance scores, we normalized the scores such that all scores summed to one. So if all variables were equally important, then 0.0023 would be the significance score for each variable. The code snippet below shows the most important variables, in order.

```
[('tGravityAcc-mean()-X', 0.03930005792390602),
 ('tGravityAcc-energy()-X', 0.03633463297803073),
 ('tGravityAcc-max()-X', 0.03615632866535244),
 ('tGravityAcc-min()-X', 0.034258135669198386),
 ('angle(X,gravityMean)', 0.03316345199109367),
 ('angle(Y,gravityMean)', 0.024350964233609905),
 ('tGravityAcc-min()-Y', 0.022530946813065843),
 ('tGravityAcc-mean()-Y', 0.020952984006897483),
 ('tGravityAcc-max()-Y', 0.02041814466337892),
 ('tGravityAcc-energy()-Y', 0.015484047653672176),
 ('fBodyAccMag-mad()', 0.012539522511055346),
 ('fBodyAccMag-energy()', 0.012064719951249374),
 ('fBodyAcc-energy()-X', 0.011605493274223609),
 ('tBodyAcc-max()-X', 0.011576515010620304),
```

Multinomial Logistic Model

While our random forest model served as a baseline for classification accuracy, we wanted to try a more interpretable model. A logistic regression classifier seemed like the ideal choice as the coefficients of the model are easily interpretable. Because our data has six classes, we were unable to use a standard logistic model that splits on binary data, and instead, we chose to

employ a multinomial logistic regression classifier, which is capable of multiclass classification through the softmax loss function.

We began by creating a logistic pipeline to vectorize, scale, and fit our base multinomial logistic model with default parameters. Our initial training error was 67%, using ML's default regularization parameters of .3 and an elastic net parameter of .8.

$$\alpha(\lambda\|\mathbf{w}\|_1) + (1-\alpha)\left(\frac{\lambda}{2}\|\mathbf{w}\|_2^2\right), \alpha \in [0,1], \lambda \geq 0$$

In the equation above, the regularization parameter is denoted as the λ and the elastic net parameter is denoted as α. The elastic net parameter contains both L1 and L2 regularization which control for overfitting.

Given an initial training error of .67, we hypertuned the parameters in the hopes of training a better model for our data. To ensure better performance, we used cross validation to split the training and test sets into five folds in order to evaluate different values of parameters for the tuned model. We experimented with the following values of possible parameters: $\lambda \in \{.01, 0.1\}$, $\alpha \in \{0, 0.5, 1\}$, and our maxiter parameter $\in \{10, 50, 100\}$. Maxiter is a parameter that accounts the number of iterations that it takes to optimize our model, similar to stochastic gradient descent. We chose to include 0 and 1 in our elastic net parameter because these values serve to convert the logistic model into a ridge regression or lasso regression, respectively. With these parameter values, we constructed ninety possible models to determine that our champion model had parameters $\alpha = 0$ and $\lambda = 0.1$.

Through the choice of $\alpha = 0$, we employed ridge regression and drastically decreased the training error from .67 to .021. Our model performed similarly on the test set with a test error of .028, The ridge regression model met all of our criteria for a champion model with high classification accuracy, efficient run time, and ease of interpretability.

One Vs. Rest Model

Though our tuned multinomial logistic regression model performed well by all measures, we wanted to try using a logistic model with a different loss function. The one vs. rest model treats one class as the positive and the rest as negative in order to classify, meaning that for our six states of activity, a one vs. rest model only needs to train six classifiers. If multiple classifiers come back positive, the model chooses the classifier with the highest confidence. Our model performed with a slightly higher error on the testing set of .07, but with a significantly faster run time. For the purposes of our study, we chose the more accurate multinomial logistic classifier as our champion model, but as applications such as fitness trackers have become increasingly reliant on speed of classification, the tradeoff between accuracy and speed in this model might be preferred.

Model Performance

	F1 Score	Recall Score	Precision Score
Base Random Forest	0.916681	0.917026	0.917654
Random Forest with 10 PC	0.818685	0.822542	0.824142
Base Logistc Regression	0.264083	0.383213	0.233033
Hypertuned Logistic Model	0.971612	0.971703	0.971591
One vs All	0.933897	0.934015	0.933943

While we originally evaluated our models on accuracy, we wanted to examine the F1 scores, recall scores, and precision scores to obtain a more complete understanding of model performance. In the table above, the scores are computed on each class and averaged into a single measurement of the performance. Our champion model, the hypertuned logistic classifier, performed best across all metrics.

Though our models were highly effective in classifying activities, we believe that future work in a few different areas could potentially improve predictive accuracy and interpretability. In preprocessing, we would like to identify the periods of transition and explore the manual labeling, as a detailed understanding of the precise moment and movements that accompany a transition in activity could be very telling in identifying the activity. Additionally, we would like to explore more research on how outliers are classified in human activity recognition and develop a method of outlier detection to improve performance using principal components, potentially decreasing runtime for our models.

Conclusion

Our project sought to develop a model which could accurately classify the activity that a user is performing with consideration to the duration of run time and ease of interpretability. After data preprocessing, exploratory data analysis, model building and tuning, and performance evaluation, we determined that our champion model was the tuned multinomial logistic regression classifier because of its high predictive accuracy, efficient runtime, and interpretability. From our random forest model, we determined that the feature variables with the most predictive power for classification were gravity and angle of body movement. Suggestions for model improvement in future studies include further outlier detection, experimentation with the number of principal components, and an exploration of the manual labeling process.

References

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine. International Workshop of Ambient Assisted Living (IWAAL 2012). Vitoria-Gasteiz, Spain. Dec 2012

Glorfeld, L. W. (1995). An Improvement on Horn's Parallel Analysis Methodology for Selecting the Correct Number of Factors to Retain. *Educational and Psychological Measurement*, *55*(3), 377–393.

"Number of Smartphone Users Worldwide 2014-2020." *Statista*, www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/.

UCI Machine Learning Repository: Human Activity Recognition Using Smartphones Data Set, archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones.