CS229 Final Project
Human Activity Recognition using Smartphone Sensor Data

Nicholas Canova, Fjoralba Shemaj

December 2016

Abstract

This paper focuses on building classifiers that accurately identify the activities being performed by individuals using their smartphone sensor data. We review the performance of the models, and make suggestions that could improve future accuracy. Exploratory data analysis and visualization techniques are used to gain a better understanding of the way users behave and how activities differ from one another.

1 Introduction

As more sensors are being built into mobile phones to measure our movements, positioning and orientation, the opportunity to understand this data and make improvements in our daily lives increases. The scope of our project consists of analyzing mobile phone sensor data in the context of activity recognition. More specifically, our objective is to build a model that accurately classifies whether an individual is walking, walking upstairs, walking downstairs, sitting, standing or laying using sensor data.

Studying activity recognition offers several benefits and enables many new applications. Mobile health applications that track a user's activities over time can be beneficial for elderly assistance or personal health monitoring. In addition to providing personal support, this research also has connections to various fields of study including medicine, human-computer interaction, and sociology.

2 Dataset and Prior Research

2.1 Description of the dataset

We obtained our dataset from the UC Irvine Machine Learning Repository [1]. For the original construction of the dataset, an experiment was carried out with 30 participants, having each person wear a Samsung Galaxy S2 smartphone containing an accelerometer and a gyroscope, while performing the six activities mentioned above. The smartphone collected 3-axial linear acceleration and angular velocity measurements, each at a constant rate of 50 hertz, and the experiment was recorded for manual labeling of the response variables. Each individual observation in our dataset is a construction of sensor signals received over a 2.56 second interval window, or 128 readings per window, with consecutive observations overlapping by 50% in time. Feature variables for the dataset were then constructed by calculating metrics from the accelerometer signals in the time and frequency domain, including the mean, standard deviation, signal magnitude area, entropy, signal frequency, etc. In total, each observation corresponds to 561 constructed features from the data collected.

The dataset has been split into 70% training and 30% test data, with 21 of 30 participants in the train data and the remaining 9 participants in the test data. The disjoint nature of the training and test split is important to consider; an effective model at recognizing activities should be able to predict the activities of new individuals. Since each study participant walks, stands and generally performs activities with differences in his or her movements, testing the performance of the model on individuals not in the training data is critical. While a model trained and tested on the same set of individuals could perform better, this would not meet the objective of our project.

2.2 Related research

Anguita et al. [2], the team that performed the original experiment, focused on applying a support vector machine adapted for multiclass classification, using computational efficiencies that exploit fixed-point arithmetic. This computational efficiency would allow applications build using this model to perform better on smartphones, since the approach requires less memory, processor time and power consumption.

Bao et al. [3] developed algorithms to detect physical activities from everyday tasks, and observed that while some activities are classified more accurately with subject-independent training data, others require subject-specific training data. This suggests that multiple sensors aid in recognition because conjunctions in acceleration feature values can help to identify many activities.

Mannini et al. [4] analyzed activity recognition for ambulatory monitoring and pervasive computing systems, where classification of human motion is analyzed, with a focus on the computational cost employed for this purpose. The group employed naive bayes, hidden markov models and support vector machines, amongst other algorithms.

3 Data Visualization

To capture the structure of our data, and better understand the distinctions between the categories of our dataset, we implemented two well-known algorithms: principal component analysis (PCA) and t-distributed stochastic neighbor embedding analysis (t-SNE). Figure 1 displays the projection of our dataset onto a two dimensional plane using the first two principal components obtained by PCA.

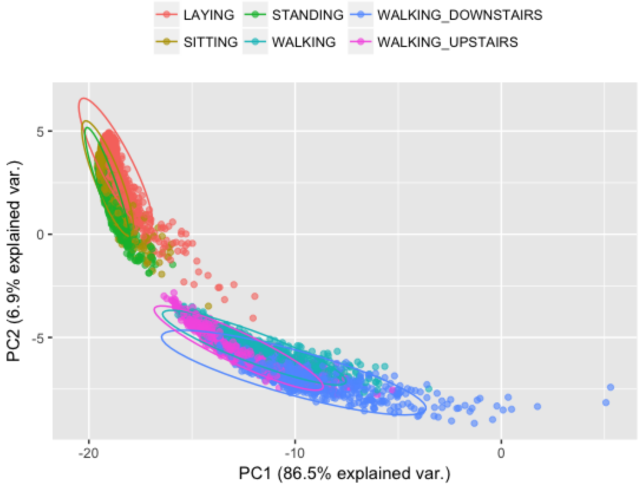


Figure 1: 2D projection of the data with PCA

Even though these two components explain a large portion of the overall variance in the data, approximately 93%, PCA can only represent the structure of the data through linear subspaces. Alternatively, the t-SNE algorithm (see Figure 2 below) can capture interesting non-linear paths and hence, looking at both types of visualization can provide useful insights from the data.

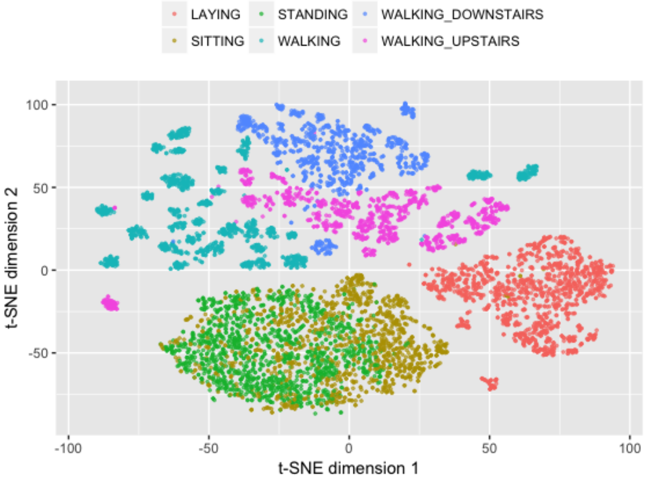


Figure 2: 2D projection of the data with t-SNE

Both algorithms effectively distinguish between activities of motion (walking, walking upstairs, walking downstairs) and static activities (sitting, laying, standing), and each of the activities are well represented by a cluster. Within all activities, sitting and standing overlap most; this is reflected in the normal ellipses overlayed for each class on the PCA plot, which display a 95% confidence region. This suggests that distinguishing these activities from one-another may pose a problem for our models.

4 Models

As mentioned above, our main objective is to construct a highly accurate classifier that generalizes well on data from new individuals. For this purpose, we have tested the performance of different classifiers, and assessed why some models performed well while others performed poorly. Algorithms implemented, as well as our motivation for each algorithm, include:

- Multinomial model [5] - One of the less complex models implemented. Given the size and high dimensionality of our data, we decided to start with a model less prone to overfitting that would serve as a baseline for the performance of more complex models.

- Support vector machines [6] - As indicated by the PCA figure, some clusters fully overlap while other clusters only partially overlap, dependent on how the corresponding activities were performed. Therefore, we would expect maximizing margins when separating these activities to result in good performance. We chose to implement SVMs using a one-vs-one approach that trains a separate classifier for each different pair of labels, as this generally outperforms a one-vs-all approach, particularly in the case of similar classes. We experimented with linear, radial-basis and polynomial kernels, tuning each model and evaluating their performance.

- Gradient boosted trees [7] - Our data is high-dimensional and there is a high level of interaction among the features, both of which boosted trees tend to handle well. We were particularly interested to see how this model would perform compared to SVMs.

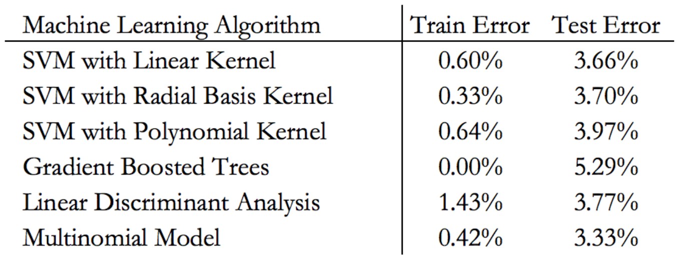
- Linear discriminant analysis [8] - The potential of the model for high accuracy was inferred from the projected data using PCA, which indicated visible clusters for each activity.

The parameters of each model require a certain amount of tuning and experimentation to optimize performance. Tuning for each of the models has been performed exclusively on the training data via 7-fold cross-validation, splitting the training data into disjoint training and validation sets, while the test data is held out solely for a final performance analysis.

5 Results

5.1 General results

Our dataset contains roughly an equal number of observations for each of the six activities. Additionally, while specific applications of activity recognition may require that one or more activities be more accurately classified than others, given our general analysis we chose to weight each activity equally. As a result, we use the overall misclassification rate on the test data as our primary performance metric. The train and test errors for each of our analyses are displayed below:



Machine Learning Algorithm	Train Error	Test Error
SVM with Linear Kernel	0.60%	3.66%
SVM with Radial Basis Kernel	0.33%	3.70%
SVM with Polynomial Kernel	0.64%	3.97%
Gradient Boosted Trees	0.00%	5.29%
Linear Discriminant Analysis	1.43%	3.77%
Multinomial Model	0.42%	3.33%

Figure 3: Misclassification rates by model

Each model displays similar performance, other than gradient boosted trees which had a higher misclassification rate. The similarity of the test errors suggests that increasing the complexity of the model does not necessarily improve its performance. In general, models with linear decision boundaries (LDA, multinomial, and linear kernel SVM) did perform slightly better than gradient boosted trees and SVMs with radial-basis kernel and polynomial kernel of degree two.

From visualizations of the projected data, we can expect fitting models with linear boundaries to perform well in separating the clusters, even though the data is not entirely separable. Projecting the data onto a higher dimensional subspace to better separate the classes has clearly failed to deliver better results. One reason could be that the data cannot be perfectly separated, even when projected in higher dimensions. Secondly, models that implement linear boundaries are less prone to overfitting than models such as radial kernel or polynomial kernel SVMs and gradient boosted trees, and hence are able to generalize better. In particular, gradient boosted trees had a training error of 0%, which suggests that the model had overfit the training data, despite efforts to regularize the model by tuning the learning rate, number of iterations and tree size.

5.2 Performance of linear kernel SVM

Since the linear kernel SVM has a low misclassification rate and is computationally efficient to train, we decided to further diagnose its performance. For the purpose of feature selection, we applied PCA and experimented with training the model on a different number of principal components. The best result was obtained using the first 300 principal components, however this resulted in the same performance as simply applying linear kernel SVM to the original data. Since reducing the number of features did not improve the performance of the model, we chose to retain all 561 features. We then observed its training and test error, while varying the number of examples in the training data. The results are displayed in Figure 4 below:

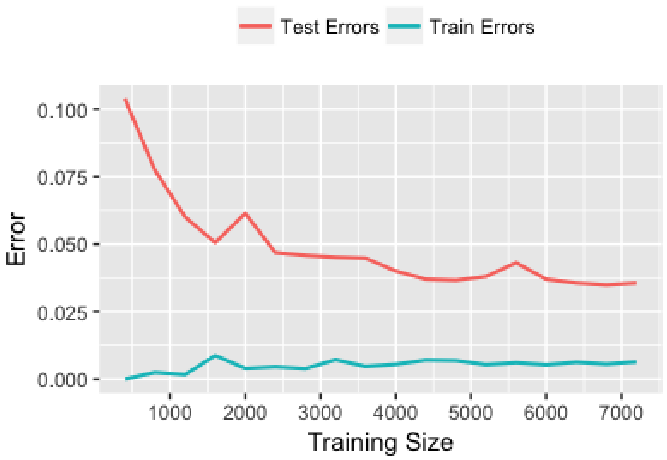
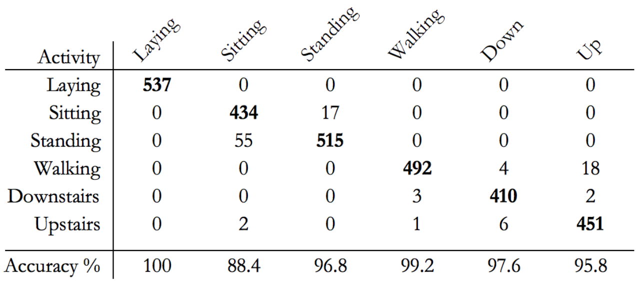


Figure 4: Test vs. Train Errors

It is clear that the two lines are converging neither too close nor too far apart from each other as the number of training examples increases. This indicates that that there is no bias or variance issue with the model. Next, to examine its accuracy in classifying each activity, we computed the confusion matrix when trained and tested on the full train and test data:



Activity	Laying	Sitting	Standing	Walking	Down	Up
Laying	**537**	0	0	0	0	0
Sitting	0	**434**	17	0	0	0
Standing	0	55	**515**	0	0	0
Walking	0	0	0	**492**	4	18
Downstairs	0	0	0	3	**410**	2
Upstairs	0	2	0	1	6	**451**
Accuracy %	100	88.4	96.8	99.2	97.6	95.8

Figure 5: Confusion matrix for SVM with linear kernel

The activity misclassified most often is sitting, which has a misclassification rate of 11.6%, with almost all errors being incorrectly identified as standing. As expected, activities of motion are more likely to be mistaken with other activities of motion, and vice versa for static activities. In addition, after examining the specific observations for which sitting was misclassified, we observed that the errors mainly occurred during the transition from standing to sitting.

5.3 Performance of specific individuals

Motivated by the idea that the model may perform differently when tested on separate individuals, we then performed a leave-one-out cross validation where we train the model on 29 users and test on the observations of the 30th user. The results are displayed below:

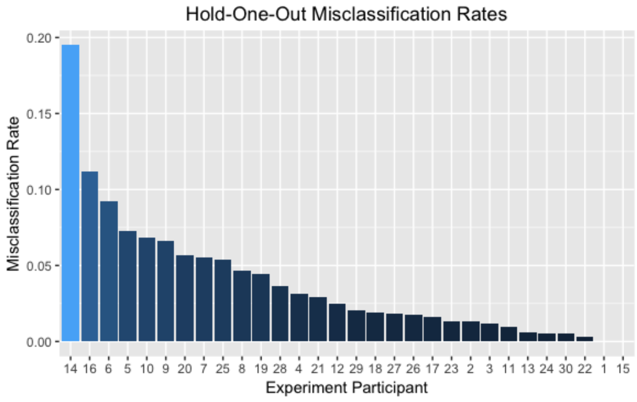


Figure 6: Misclassification rates by user

As anticipated, the misclassification rate by user ranged significantly, from 0.0% to 19.5%. Examining the individual confusion matrices, we observed for the users with the highest error rates, that one inaccurate activity generally accounted for all errors for that user. This motivates us to inspect the variability between users within each activity.

To examine whether there is a large variance between individuals, we have reduced the earlier t-SNE figure to specific activities, distinguishing by color each of the different individuals performing the activity. The two plots below correspond to the t-SNE output for standing and walking, respectively.

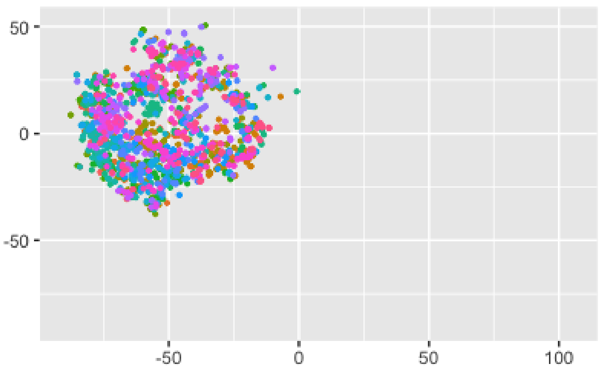


Figure 7: t-SNE plot for standing, all users

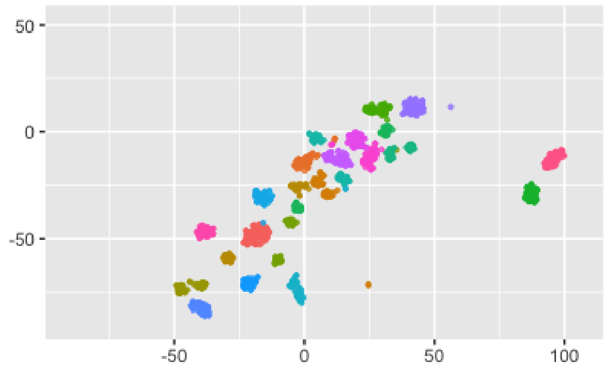


Figure 8: t-SNE plot for walking, all users

The t-SNE plot for walking shows clear variability between individuals, with each individual belonging to a noticeable cluster. On the other hand, all individuals are grouped together with respect to standing, indicative that individuals generally stand in the same manner as one-another, but have differences in the way they walk. This behavior generalizes to other activities of motion and static activities as well. This supports the research by Bao et al. that some activities are classified more accurately with subject-independent training data, while others require subject-specific training data; static activities are likely to be classified equally well using either subject-dependent or subject-independent data, while activities of motion may require subject-specific data to achieve higher accuracy.

6 Conclusion

Overall, our list of classifiers achieved relatively high performance. While the various models displayed similar test errors, the accuracy for individual users and specific activities did vary significantly. Sitting was the most difficult activity to classify, often being misclassified as standing, and perhaps having additional features to distinguish sitting from standing could help in this aspect.

Since the linear kernel SVM had a higher misclassification rate when an individual was transitioning from standing to sitting, a model that captures the time dependency in the data, such as a hidden markov model, could be useful in this case. However, since the activities in the experiment occurred in a predefined order, a new dataset where the order of activities performed varied between users, and were consistent with how users generally transition between these activities, would be necessary when implementing a hidden markov model.

7 References

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. *A Public Domain Dataset for Human Activity Recognition Using Smartphones.* 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.

[2] Anguita, Davide, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz. *Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine.* Vitoria-Gasteiz, Spain: International Workshop.

[3] Bao, Ling and Stephen S. Intille. *Activity recognition from user-annotated acceleration data*, 2004.

[4] Mannini, A., Sabatini, A.M.: *Machine learning methods for classifying human physical activity from on-body accelerometers.* Sensors 10(2) (2010) 11541175

[5] Jerome Friedman, Trevor Hastie, Robert Tibshirani (2010). *Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software*, 33(1), 1-22. URL http://www.jstatsoft.org/v33/i01/.

[6] David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel and Friedrich Leisch (2015). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071)*, TU Wien. R package version 1.6-7. https://CRAN.R-project.org/package=e1071

[7] Greg Ridgeway with contributions from others (2015). gbm: Generalized Boosted Regression Models. R package version 2.1.1. https://CRAN.R-project.org/package=gbm

[8] Venables, W. N. Ripley, B. D. (2002) *Modern Applied Statistics with S. Fourth Edition.* Springer, New York. ISBN 0-387-95457-0