Predicting Mode of Transport from iPhone Accelerometer Data

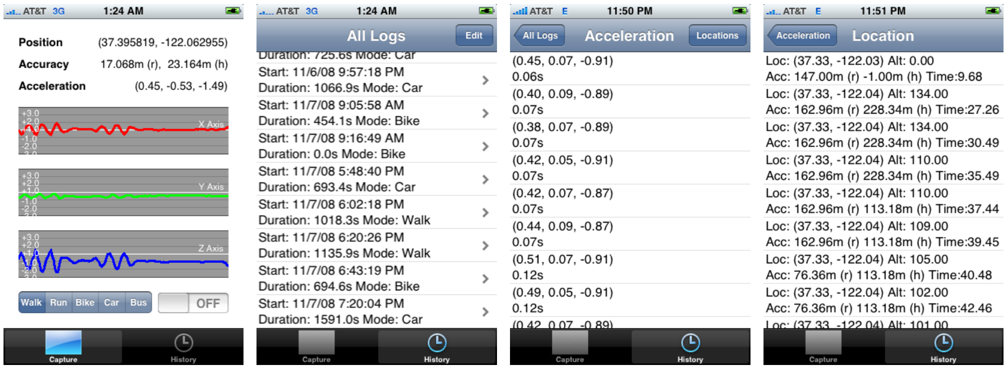
Ben Nham, Kanya Siangliulue, and Serena Yeung

Introduction

In our project, we present a method for offline classification of transportation modes from an iPhone accelerometer. Our research may be useful for accurately predicting travel times, where automatically detecting transport mode can be used as an input into a particle filter, or for automatically classifying and logging physical activity over the course of a day as part of a fitness program. While there has been research into using multiple accelerometers to classify physical activity [1, 2, 3], we are unaware of research into classifying transportation modes using a widely-available retail device such as the iPhone.

We start by extracting various features from the raw accelerometer data. We then train several learning algorithms on subsets of these features to generate a classifier that can reasonably distinguish between the different transportation modes from the acceleration stream.

Data collection



Position (37.395819, -122.062955)
Accuracy 17.068m (r), 23.164m (h)
Acceleration (0.45, -0.53, -1.49)

Duration: 725.6s Mode: Car
Start: 11/6/08 9:57:18 PM
Duration: 1066.9s Mode: Car
Start: 11/7/08 9:05:58 AM
Duration: 454.1s Mode: Bike
Start: 11/7/08 9:16:49 AM
Duration: 0.0s Mode: Bike
Start: 11/7/08 5:48:40 PM
Duration: 693.4s Mode: Car
Start: 11/7/08 6:02:18 PM
Duration: 1018.3s Mode: Walk
Start: 11/7/08 6:20:26 PM
Duration: 1135.9s Mode: Walk
Start: 11/7/08 6:43:19 PM
Duration: 694.6s Mode: Bike
Start: 11/7/08 7:20:04 PM
Duration: 1591.0s Mode: Car

(0.45, 0.07, -0.91)
0.06s
(0.40, 0.09, -0.89)
0.07s
(0.38, 0.07, -0.89)
0.07s
(0.42, 0.05, -0.91)
0.07s
(0.42, 0.07, -0.87)
0.07s
(0.44, 0.09, -0.87)
0.07s
(0.51, 0.07, -0.91)
0.12s
(0.49, 0.05, -0.91)
0.12s
(0.42, 0.07, -0.89)

Loc: (37.33, -122.03) Alt: 0.00
Acc: 147.00m (r) -1.00m (h) Time:9.68
Loc: (37.33, -122.04) Alt: 134.00
Acc: 162.96m (r) 228.34m (h) Time:27.26
Loc: (37.33, -122.04) Alt: 134.00
Acc: 162.96m (r) 228.34m (h) Time:30.49
Loc: (37.33, -122.04) Alt: 110.00
Acc: 162.96m (r) 228.34m (h) Time:35.49
Loc: (37.33, -122.04) Alt: 110.00
Acc: 162.96m (r) 113.18m (h) Time:37.44
Loc: (37.33, -122.04) Alt: 109.00
Acc: 162.96m (r) 113.18m (h) Time:39.45
Loc: (37.33, -122.04) Alt: 105.00
Acc: 76.36m (r) 113.18m (h) Time:40.48
Loc: (37.33, -122.04) Alt: 102.00
Acc: 76.36m (r) 113.18m (h) Time:42.46
Loc: (37.33, -122.04) Alt: 101.00

Figure 1. Accelerometer logging application

To collect accelerometer data, we wrote a custom application to sample the iPhone's three axis, +/- 2g accelerometer at 50 Hz while annotating the current transportation mode, as shown in Figure 1. The device was placed in the pocket of the non-dominant hip of all participants to allow acceleration magnitudes to be compared across subjects. To simulate real usage of the device, we placed no constraint on the orientation of the device in the pocket while logging data.

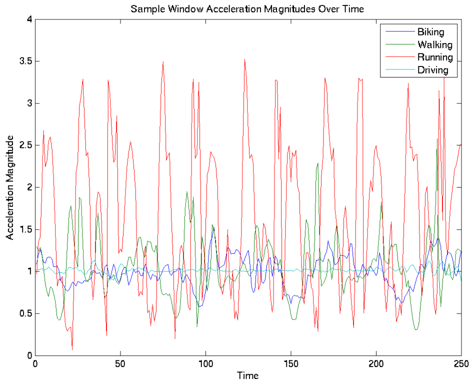
In total, four participants gathered 8.87 hours of annotated acceleration data while walking, biking, running, driving, riding trains, and riding buses. The first 70% of the dataset for each of the transportation modes was chosen as the training set, while the remaining data was used as the test set. The datasets can be found on the project website [4].

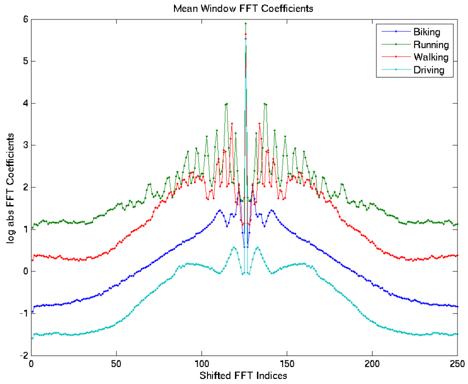
Feature Extraction

In past projects involving activity detection from accelerometer data, researchers mounted their accelerometers (and other sensors) in fixed orientations at fixed spots on all participants. We decided to have participants place their devices in their hip pockets, as this seemed like a natural location for storing a phone. However, we decided to relax the constraint that the phone be placed in a particular orientation (face up or face down and rotated portrait or landscape) to better simulate real usage of a phone; a subject would be unlikely to remember to place in their pocket in exactly the same orientation every time they finish using it.

This made correlating axis information across samples difficult, as gravity could be pointing along different axes depending on the orientation of the device in a given sample. We decided to solve this problem by first preprocessing our data by collapsing the triaxial acceleration data over time into a vector of acceleration magnitudes. This solved our orientation problem, but meant we could not directly use previous research the trained learning algorithms on triaxial data [4].

We then split the magnitudes into windows of 5 seconds (250 samples) each, with each window overlapping the previous window by 50%, as previous studies found this to be an effective window size [5]. A 250-point FFT was taken on each window, and the magnitude of the Fourier coefficients stored in the final training matrix.

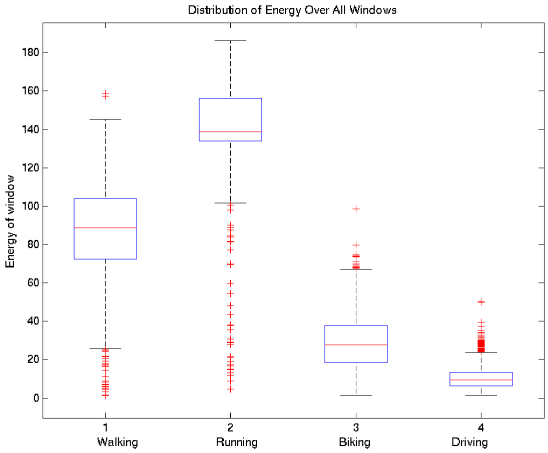


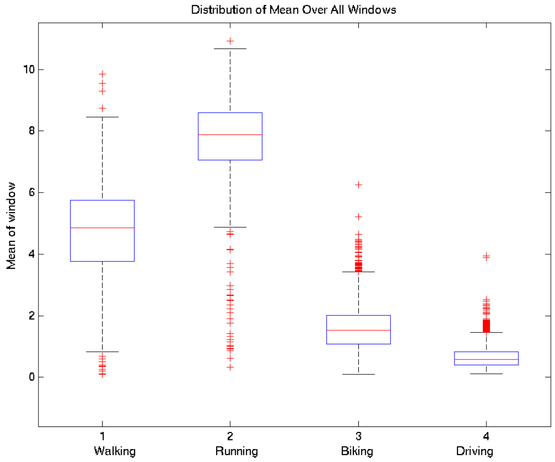


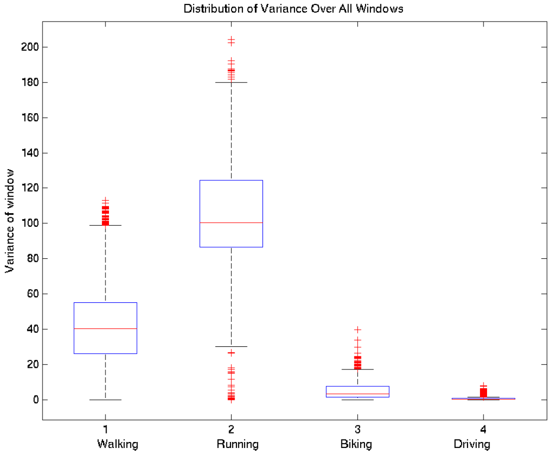
The figures above show a sample magnitude window, along with the mean of FFT components across all windows. From the FFT plot, we can see that, at least on average, there are distinct frequency characteristics for each transportation mode. Running shows the highest peaks (corresponding to a higher energy signal) and a higher fundamental frequency than walking, which makes sense since the pace and impact of running is generally higher than with walking. Both running and walking show many frequency peaks, probably from the multitude of vibrations that occur when a shoe hits the ground with some force. On the other hand, biking and driving show much lower peaks, as there is not the same style of impact in these activities; indeed, when driving, subjects usually stay stationary, and even when a car is accelerating or decelerating, the acceleration is generally is quite smooth.

From these FFT coefficients, we extracted the following 253 features:

- The magnitudes of the 250 FFT components
- Energy of the signal (squared sum of FFT components, from Parseval's Theorem)
- Mean of the signal
- Variance of the signal





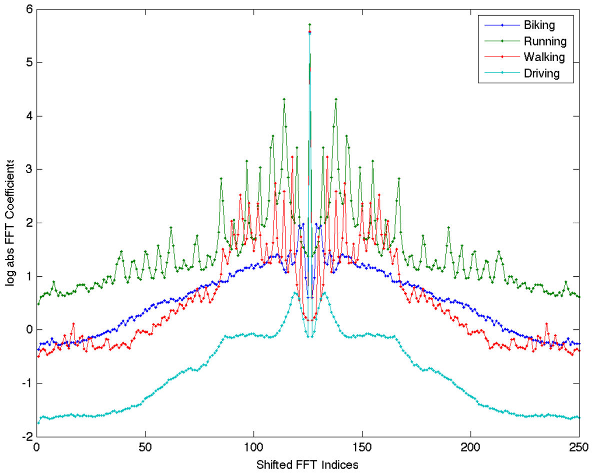


We chose to additionally extract energy, mean, and variance from the signal as previous research has showed that they were significant features when using triaxial data [5]. The figures below show the distribution of energy, mean, and variance are shown above. As in the FFT plot, running shows the most energy, mean signal strength, and variance, followed by walking, biking, and finally driving.

GDA Results

Walking	Running	Biking	Driving	Total
94.68%	98.08%	45.04%	58.40%	74.62%

Table 1. GDA Classification Results



Initially, we used GDA to model the training data as a 250-dimensional Gaussian with unique mean and covariance over the 250 FFT coefficients for each activity. When data from all three training subjects were lumped together and hold-out cross validation was performed, the subject-independent test results were as shown in Table 1. Walking and running windows were classified correctly with reasonably high accuracy, but driving and especially biking had much lower accuracy. After re-running hold-out cross validation on each subject's data samples independently, we found that GDA predicted Ben's motion correctly 88.26% of time, Pao's motion correctly 96.88% of the time, and Serena's motion correctly 58.40% of the time. The poor classification performance on Serena was caused by most of her biking windows being misclassified as walking, which also brought down the overall accuracy of the algorithm. This made sense upon examining Serena's average FFT window above, which shows significant overlap between the coefficients for biking and driving.

SVM Results

Since most of the error in the GDA model was due to its inability to separate out the differences between sets of data that were also hard to separate by eye, we decided to try an SVM with a higher dimension feature space (by using a Gaussian kernel) to try to better separate the data. We used libsvm with one-vs-one max-wins voting for multi-class support, 5-fold cross validation, and grid search through the parameter space (the constant cost c and the

Gaussian parameter gamma) to achieve the results shown in Table 2. In particular, Serena's test accuracy improved to a comparable level of accuracy as Ben and Pao, so there were no longer specific classification tests that resulted in abnormally high error.

	250 Components		162 Components		3 components	
	Train	Test	Train	Test	Train	Test
Ben	98.53%	93.24%	95.66%	97.14%	92.65%	94.35%
Pao	97.73%	92.78%	93.09%	91.88%	85.09%	87.83%
Serena	99.81%	92.38%	98.21%	88.65%	95.16%	90.69%
All	98.01%	93.88%	94.05%	*	84.17%	*

Table 2. SVM Classification Results

A confusion matrix for training and predicting on data from all subjects on all parameters is shown below. The input test classes are shown on the rows, while the predicted classes are shown in the columns. Of all pairs of activities, biking and driving were confused the most. From the energy, mean, and variance graphs in the Feature Extraction section, this seems reasonable since biking and driving are similarly low acceleration-energy activities.

	Walking	Running	Biking	Driving
Walking	922	6	4	0
Running	2	578	1	1
Biking	20	4	703	25
Driving	7	0	165	1402

Table 3. Confusion Matrix When Classifying on All Component

The difference between training accuracy and testing accuracy in Table 2 when training on all 250 FFT components showed that the SVM model was likely suffering from high variance and overfitting. The training accuracy was substantially higher than the testing accuracy for the feature set of 250 components, and notably, Serena's originally low training accuracy (from the GDA) was now fitted to an almost perfect 99.81%, although testing accuracy was only 92.38%. This suggested that either more data or better feature selection might improve the model. Adding more training data was not achievable given the time constraints, so we focused more on feature selection. In addition to the 250 frequency components, we added 3 more features--energy, mean, and variance--which were found to be good predictors in previous studies that used more sensor data [5]. We then used a libsvm library that used an F-score heuristic to choose between the 253 total parameters. The algorithm chose 162 paramters from the 253, and the results are from training on these parameters are shown in Table 2. We also tried to train on only the energy, mean, and variance parameters to see if that would give "good enough" results so that a classifier could be trained efficiently in real-time. We had trouble with libsvm infinite looping on several configurations indicated by the asterisk in the table.

Choosing fewer parameters did narrow the gap between training and testing accuracy for Ben and Pao, but it did not work well for Serena and did not result in a significant improvement in accuracy. This may have been in part due to the nature of the 250 components as essentially all part of a larger frequency category, so that there were only 4 completely separate feature categories to start with. Also, there were only 3 subjects with different gaits and data distribution characteristics (i.e. different frequency peaks), making it difficult to pick out the most important frequency components from only these 3 distributions. Sampling more subjects, so that there would be a more normal distribution of gait and frequency characteristics, might have resulted in more informative feature selection, and a consideration of other feature types (i.e. frequency peaks) might have made feature selection more effective, but we did not have the time to explore these aspects in the time allotted.

Future Work

In this project, we were able to achieve reasonable accuracy training and testing on the same subjects, but we did not look much into training on one set of subjects and testing on an unrelated subject. Developing such a general classifier would likely require significantly more data collected from a larger distribution of subjects. For future work, we think it would be both interesting and useful to look into this problem, as it could be used to detect activities out-of-the-box in a consumer product. We also believe that collecting additional data and analyzing features more closely could help further improve the accuracy of our model, as discussed above.

Acknowledgements

Morgan Quigley of Stanford helped us greatly by giving us the initial project idea, as well as mentoring us throughout the project.

References

1. Bao, L., Intille, S.: Activity Recognition from User-Annotated Acceleration Data. In: Proc.Proc. Pervasive (2004) 1-172.
2. E. Munguia Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman, "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor " in Proceedings of the International Symposium on Wearable Computers: IEEE Press, 2007.
3. J. Lester, T. Choudhury and G. Borriello, "A Practical Approach to Recognizing Physical Activity," Proc. 4th Int"l Conf. Pervasive Computing (Pervasive 06), LNCS 3968, Springer, 2006, pp. 1–16.
4. <http://svn.nhaminated.com/viewvc.cgi/project/?root=cs229
5. N. Ravi, N. Dandekar, P. Mysore, and M. Littman, "Activity Recognition from Accelerometer Data" in AAAI: AAAI Press, 2005.