Near Real-Time Transportation Mode Detection Based on Accelerometer Readings

Jasna Urbančič, Luka Bradeško, Matej Senožetnik
Jožef Stefan Institute
and
Jožef Stefan International Postgraduate School
Jamova cesta 39
1000 Ljubljana, Slovenia

ABSTRACT

This paper describes a method for automatic transportation mode detection based on smartphone sensors. Our approach is designed to work in real-time as it only requires 5s of sensor readings for the detection. Because we used accelerometer instead of GPS signal it uses less battery power and is therefore more user and phone-friendly. For the mode detection we use multiple support vector machine models which enable us distinguishing between multiple modes (bus, train, car). Before the classification, raw measurements are preprocessed in order to cancel out the constant acceleration that is caused by the force of gravity. The results of the paper are promising and are based on the collected training data from approximately 20 hours of driving on trains and public buses in Ljubljana.

Keywords

activity recognition, support vector machine classification, accelerometer

1. INTRODUCTION

Nowadays most smartphones have built-in sensors that measure motion, acceleration, orientation, and various other environmental conditions with quite high precision and sampling frequency. This can be used with great success in everyday challenges, for example tracking and routing applications. It has been proven that smartphone sensors are useful in monitoring three-dimensional device movement and positioning [1], and also user's activity detection, which is also in the domain of this paper.

Mobile operating system developers are aware of such applications, therefore their APIs include activity recognition packages. However, they detect only a few modes - still, walking, running, cycling, and in vehicle. Such coarse-grained classification is not enough for tracking and routing purposes, specially in use-cases for urban environments, where public transportation with buses and trains can be a good alternative to private vehicles.

As the main smartphone APIs already support fine-grained classification of non-motorized forms of transportation [2, 3], we focused on distinguishing means of motorized transport, specifically cars and buses, as the majority of passenger traffic in Ljubljana represent cars and buses. Trains and motorbikes are not that common, whereas subway and tram infrastructures do not exist. Our goal is to recognize each mode of transportation in near real time while mobile phone users are traveling.

2. RELATED WORK

Ever since smartphones appeared and gained accessibility there has been a lot of research activity for their usage in user activity recognition and transportation mode detection. While the first attempts to recognize user activity were done before smartphones, the real effort in that direction started with the development of mobile phones having built-in sensors [7]. Besides GPS sensors, also GSM triangulation and local area wireless technology (Wi-Fi) can be employed for the purpose of transportation mode detection. However its accuracy is relatively low compared to GPS, therefore we deem these out of scope of this paper[8].

Latest state of the art research is focused on transportation mode detection based on GPS signal and/or accelerometer data. Approaches that rely solely on GPS trajectories require GPS signal of high-quality, whereas phones GPS receiver is generally severely shielded during daily activities [10]. This may occur during travel underground, inside stations, or even when user is not sufficiently close to a window when traveling in a vehicle [5], and results in loss of positioning information. Another known issue when using GPS signal on mobile device is high power consumption [5], which is especially not pleasing in the case of longer commutes. Both of these two issues suggest that the accelerometer sensor is more appropriate for activity detection.

Another advantage of using accelerometer data over GPS signal is that it does not require additional external data. Many researches using GPS data used external data, such as GIS data on bus stations, bus routes and railway lines [9].

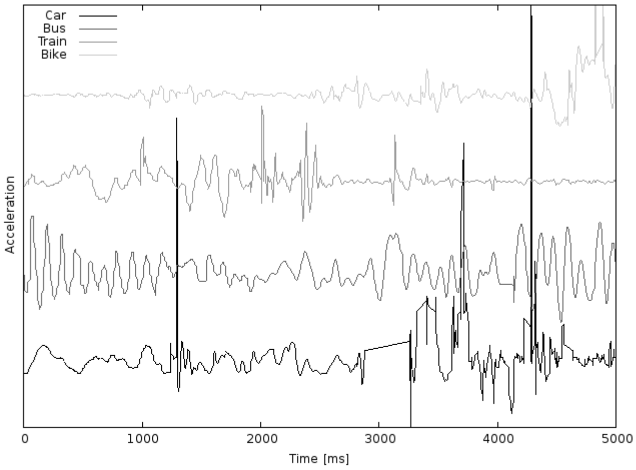

Figure 1: Amplitudes of raw accelerometer data for different means of motorized transportation.

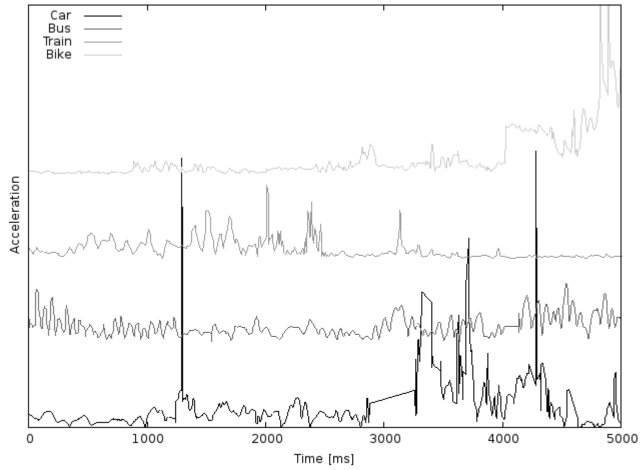

Figure 2: Amplitudes of preprocessed accelerometer data for different means of motorized transportation.

3. TRANSPORTATION MODE DETECTION

For the purpose of collecting accelerometer data we extended the GPS tracking mobile application with the accelerometer measuring ability. The phone sensor measures acceleration forces in m/s^2 for all of the three physical axes (x, y, z). The sampling rate is 100Hz (1 measurement every 10 ms). To increase the diversity of the training data-set, measurements were acquired in multiple ways:

- Person is collecting data while traveling by the car and stops the collection at the destination.
- Person is collecting data while traveling by the bus and stops the collection on exit.
- Person is traveling by the train and is collecting data until the arrival to the final destination
- Person is collecting data while driving a motorbike and stops the collection at the destination.

We collected approximately 20 hours of travelling measurements, with the travel modes distributed as follows: $car-57\%$, $bus-32\%$, $train-11\%$, $motorbike-0.1\%$. Amplitudes of the raw accelerometer data are shown in Figure 1.

3.1 Preprocessing

In order to reduce the computation time and to have faster response time for real-time classification, we split the recorded accelerometer signal into smaller pieces that do not exceed 5s timewise. This enables us to work on chunks of record that span only through 5s or less. This additionally preserves battery life, saves space, and reduces usage of mobile data.

Acceleration measurements are correlated with the orientation of the phone in 3D space, as gravity is measured together with the dynamic acceleration caused by phone movements. Thus we have to be able to separate the constant acceleration caused by the gravity and the dynamic part of the acceleration.

The gravity estimation algorithm works as follows: for a chosen sampling interval (in our case 1s), obtain an estimate of the gravity component on each axis by averaging all measurements in the interval on that axis [6]. After obtaining the estimates, we subtracted the gravity component from all of the entries on corresponding axis in given time interval. Through this we obtained only the dynamic acceleration component of the signal. Amplitudes of dynamic accelerometer signal are shown in Figure 2.

3.2 Classification

After preprocessing the accelerometer readings we extracted features for the classification process. We used mean, variance, skewness, 5th, and 95th percentile of acceleration data on all three axes. We also split the acceleration into positive part, which indicates that the velocity of movement in that direction is increasing, and negative part, which indicates that the velocity is decreasing, and calculated the same statistics on these two parts.

We used support vector machine (SVM) classifier as it was previously successfully used in similar work [8]. The implementation was SVM classifier (SVC) from QMiner package. QMiner is an open source analytics platform for performing large scale data analysis written in C++ and exposed via a Javascript API [4].

First we focused on the binary classification of car and bus transportation versus the rest. We trained binary classifiers for each of the labels in one against the rest manner. That means that examples labled with this particular label represented positive examples, whereas all other examples, regardless of class represented negative examples. However, we also did one against one classification for each pair of labels. That means that examples of one class were marked as positive, examples of another class were marked as negative, and the rest of the learning set was filtered out. Later, we extended this to support multi-class classification. For multi-class classification we used binary models and combined their predictions based on the distance between the

Table 1: Table of all extracted features.

Acceleration data	Features
X axis	Mean (Total, Acceleration, Deceleration)
Y axis	Standard deviation (Total, Acceleration, Deceleration)
Z axis	Skewness (Total, Acceleration, Deceleration)
Amplitude	5th percentile (Total, Acceleration, Deceleration)
	95th percentile (Total, Acceleration, Deceleration)
Total number of features	60

Table 2: Classification accuracy, precision, recall, and F1 score for binary classifiers of different transportation modes as results of 10-fold cross validation.

	Accuracy	Precision	Recall	F1 score
Car	0.855	0.852	0.910	0.880
Bus	0.720	0.620	0.694	0.655
Train	0.876	0.726	0.671	0.697

separating hyper plane and the test sample.

4. EVALUATION

Evaluation section is divided into two parts. In the first one are presented the results of experiments with one against the rest classification, whereas in the second part we discuss the results of one against one classification. In both parts we considered two different scenarios. In the first one we tested if our approach can recognize a specific transportation mode from all the others (one-vs-all) or if SVC can distinguish between two specific modes of transportation (one-vs-one). In the second, we used previously obtained binary models to classify to three(3) different classes. Our main focus was recognizing traveling by cars and buses as these represent majority of the passenger traffic in Ljubljana and therefore the majority of our training data.

We measured performance of the models with classification accuracy, precision and recall. Furthermore, we estimated a harmonic mean of precision and recall with the F1 score. We used 10-fold cross validation to tune the parameters of each binary model.

4.1 One against all

Binary classification (one-vs-all) was done for each of the three classes (car, bus and train). Classification accuracy, precision and recall for the most suitable values of parameters are listed in Table 2.

We got the best results (accuracy, precision and recall) for car travel detection. There was some drop in the performance of bus travel detection, and even bigger drop for the train detection. We assume that the main cause for performance drop is smaller training data set for bus and train. We plan to resolve this with additional data-set collection as part of the future work.

For the multi-class classification we used binary models from Table 2. We mapped results into 4 classes (car, bus, train and UC - unable to classify). If according to the binary classification, an instance belongs to none of the classes or more than one, we label it as UC (unable to classify). The plan behind UC is, that we ask the application providing accelerometer data for new sample, which can help us reclassify into the proper class. The results in Table 3 and Table 4 are not surprising as the majority of cars is classified as cars and also most of misclassified cars are instances that belong to either none or more than one class. In contrast to cars, proportions of correctly classified buses and trains are smaller than the proportions of UC for these two classes, which shows that our approach to combining predictions for multiple classifiers might not be the best.

Table 3: Confusion matrix for classification for 3 classes with 10-fold cross-validation.

True\ Pred.	Car	Bus	Train	UC
Car	0.818	0.012	0.008	0.162
Bus	0.198	0.219	0.072	0.511
Train	0.118	0.042	0.344	0.496

Table 4: Classification accuracy, precision, recall, and F1 score for classification with 3 classifiers with 10-fold cross-validation.

	Accuracy	Precision	Recall	F1 score
Car	0.823	0.826	0.818	0.827
Bus	0.725	0.844	0.219	0.347
Train	0.859	0.683	0.181	0.286
Average	0.803	0.784	0.401	0.535

4.2 One against one

We did similarly for one-vs-one binary classification. Results of this are shown in Table 5, which shows that cars are very well distinguishable from trains and vice versa. Buses are less distinguishable from cars and trains, however the accuracy and F1 score of all the classifications are still above 0.8.

We used these six binary classifiers for multi-class classification. Confusion matrix and accuracy, precision, recall and F1 score are listed in Tables 6 and 7. Tables show that classification accuracy, precision, recall and F1 score are higher than in case of one against all classification. Confusion ma-

Table 5: Accuracy / F1 score for one-vs-one binary classification. Rows represent positive examples, whereas columns are negative examples.

	Car	Bus	Train
Car		0.848/0.889	0.943/0.962
Bus	0.856/0.832		0.858/0.896
Train	0.934/0.883	0.815/0.760	

Table 6: Confusion matrix for classification for 3 classes with 10-fold cross-validation.

True\ Pred.	Car	Bus	Train
Car	0.893	0.088	0.019
Bus	0.282	0.573	0.145
Train	0.167	0.162	0.671

Table 7: Classification accuracy, precision, recall, and F1 score with 10-fold cross-validation.

	Accuracy	Precision	Recall	F1 score
Car	0.825	0.785	0.893	0.835
Bus	0.787	0.724	0.573	0.640
Train	0.886	0.672	0.671	0.671
Average	0.832	0.727	0.712	0.716

trix shows that nearly 90% of cars is classified correctly, whereas for buses and trains the percentage of correctly classified instances drops to 57% and 67% respectively. It also shows that trains are equally likely to be miss-classified as cars and the buses. In contrast to trains, buses are more often miss-classified as cars than trains. In comparison with one-vs-all multi-class classification, values of recall and F1 score are much higher, whereas accuracy and precision for these two approaches are comparable.

5. CONCLUSION

In the presented work we showed that it is possible to detect transportation mode using support vector machine, with short readings of accelerometer signal. This proves that near real-time activity detection with fine-grained motorized transportation modality is possible.

Nonetheless, there are still some issues for the future work. First one is regarding unbalanced data set as we will have to collect more data, especially for train and motorbike detection. Secondly, our task will be improving binary classification regarding buses as this class was most often misclassified. However, this might be also caused by our policy for combining multiple binary classification results, which is also something we will have to work on.

6. ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and the ICT program of the EC under project OPTIMUM (H2020-MG-636160).

7. REFERENCES

[1] Sensors overview. `https://developer.android.com/guide/topics/sensors/sensors_overview.html`, 2016. [Online; accessed 25-August-2016].

[2] `ActivityRecognition`. `https://developers.google.com/android/reference/com/google/android/gms/location/ActivityRecognition`, 2016. [Online; accessed 25-August-2016].

[3] `CMMotionActivity`. `https://developer.apple.com/library/ios/documentation/CoreMotion/Reference/CMMotionActivity_class/index.html#//apple_ref/occ/cl/CMMotionActivity`, 2016. [Online; accessed 25-August-2016].

[4] B. Fortuna, J. Rupnik, J. Brank, C. Fortuna, V. Jovanoski, and M. Karlovcec. Qminer: Data analytics platform for processing streams of structured and unstructured data. In *Software Engineering for Machine Learning Workshop, Neural Information Processing Systems*, 2014.

[5] S. Hemminki, P. Nurmi, and S. Tarkoma. Accelerometer-based transportation mode detection on smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 13. ACM, 2013.

[6] D. Mizell. Using gravity to estimate accelerometer orientation. In *Proc. 7th IEEE Int. Symposium on Wearable Computers (ISWC 2003)*, page 252. Citeseer, 2003.

[7] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)*, 6(2):13, 2010.

[8] M. A. Shafique and E. Hato. Use of acceleration data for transportation mode prediction. *Transportation*, 42(1):163–188, 2015.

[9] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu. Transportation mode detection using mobile phones and gis information. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 54–63. ACM, 2011.

[10] P. Widhalm, P. Nitsche, and N. Brändie. Transport mode detection with realistic smartphone sensor data. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 573–576. IEEE, 2012.