Detecting Time Interval Patterns from Smart Card Data

Paul Bouman Evelien van der Hurk Leo Kroon Ting Li Peter Vervest

Rotterdam School of Management, Erasmus University.
Burgemeester Oudlaan 50, 3062 PA, Rotterdam, The Netherlands.

Abstract

During the past decades, the modelling of transport demand by activity based methods has gained considerable attention from the scientific community. Such demand models offer a greater modelling flexibility than traditional models, by modelling transport demand as a phenomenon which emerges from the desire to perform activities at different locations, as opposed to more traditional models where an origin destination demand matrix of trips is distributed over different routes and modes.

One of the drawbacks of the activity based paradigm is that data related to activities is more difficult to collect than traffic counts. Modern technologies, such as smart card ticketing systems and location detection services in smart phones, allow us to collect more detailed accounts of the movements of individual passengers. This gives us the possibility to analyse consecutive journeys and therefore the time a passenger spends in a certain location. This information can be very useful from an activity based modelling perspective.

In this paper we take an exploratory approach to derive important activity time intervals from smart card data. We apply a clustering algorithm on the intervals observed at individual stations to detect which time intervals capture enough activities. We then construct a classification tree that allows us to classify the activities and analyse activity chains of individual passengers. We count pairs of consecutive activity classes, visualise the results as a network and calculate which triplets of consecutive activities occur most often. Using this approach, we are able to identify activity patterns that differ from the typical time windows associated with home-work activities.

1 Introduction

One of the most valuable pieces of information during the development and operational planning of passenger transportation systems is information on passenger demand. Understanding how this demand develops allows governments and public transport operators to assess whether certain infrastructure investments or changes in the service network are profitable. It also serves as input for decisions on service frequencies, the choice of vehicle types and rolling stock allocation.

Traditional demand models typically consist of an estimated origin-destination matrix of trips and a traffic assignment model assigning routes to OD-pairs in a possibly multi-modal transportation network. One of the drawbacks of this approach is that it is not very straightforward to make the matrix time dependent, introduce heterogeneous groups of travellers or to include the change in demand resulting from interaction between passengers and the network, such as for example due to crowding.

Activity Based Models [3] provide an improvement in this regard. The main idea of this paradigm is that transport demand emerges from many individual desires to perform certain activities at different locations at certain times. An example implementation of such a model is the open source agent-based transport simulation package MATSim [4], that has been applied at different locations around the world. The input required for such models consists of individual day plans that define a chain of activities. Since this input data cannot be directly deducted from an OD-matrix, random plans generated from economic and geographical data are often combined with travel diaries collected through surveys.

In this paper we develop a method to deduce and analyse activity patterns and activity sequence patterns within the time dimension. We define an activity as a combination of a time interval and a location. These activities are reconstructed from the set of trips stored in the data for a specific person. Using both clustering and classification methods, we identify important activity time intervals and analyse common activity chains. We consider a time interval to be important if it represents at least 10% of the activities at a station in the network. We are not only able to identify home-work patterns, but also identify shorter activities. Moreover, the activity chains provide information on the order of different activities. We aim to extend our method to include spatial dimensions in the future, by classifying stations into groups based on the temporal patterns outputted. We believe that the results obtained using our method can provide public transport operators insight into how their network is being used and give valuable input for activity based models.

2 Related Work

Pelletier et al. [10] present an excellent general review of smart card data research in public transport during the years 2000-2010. As this is an extensive literature review, we only present a short overview of research focused on activity analysis based on smart card data.

Agard et al. [1] analyse user behaviour by translating smart card records per card to a binary vector indicating smart card activity during four fixed time slots, defined by the operator that owned the data set. They find four main travel patterns using hierarchical clustering, the top two of which correspond to a home-work-home pattern and a home-study-home pattern. Morency et al. [9] focus on the variation in temporal patterns using a k-means clustering algorithm. They also translate journey sequences to vectors, but here a vector represents 24 hour time slots and contains binary indicators whether a passenger has boarded a vehicle during this time slot. Using clustering with the Hamming distance measure and the component wise median to derive cluster centroids, they are able to derive regularity indicators from the raw data. Devillaine et al. [7] present an analysis focused on the temporal distribution of activities based on smart card data from both Santiago, Chile and Gautineau, Canada. Their classification is based on both temporal aspects as well as card type. The assigned classes are work, study, home and other. They find that the temporal distribution of activities in Santiago differs from Gautineau. Activities classified as other have peaks at their starting times when they start more often around noon or four in the afternoon in the Gautineau network, while they are more evenly distributed in the Santiago network.

A different methodology to analyse spatio-temporal patterns is to calculate eigenbehaviors [8]. The general idea of the method is to apply Principal Components Analysis on vectors of binary variables representing time slot/location combinations. While this method is usually able to reduce a matrix of vectors to a few dominant eigenvectors, the fractional nature of the eigenvectors makes them complicated to interpret, especially if the goal is to create input for activity based models.

3 Smart Card Data

The Dutch smart card system, called "OV-Chipkaart" is a nation wide smart card for payment of public transport journeys across modes and operators. The system is operated by the common smart card authority "Trans Link Systems", which collects the transactions and provides the operators with the data of their customers. This is raw transactional data where each record contains at least the following fields: a unique media ID of the smart card, date and time of the transaction, an ID specifying the station or stop where the transaction took place and the type of the transaction (i.e. check in or check out). One of the important features of the Dutch implementation is that passengers need not only check in when they start their journey, but also need to check out at the end of their journey. As a result, we do not need to apply techniques to estimate alighting points for the different smart cards, but can just look at the check outs.

In order to analyse the intervals corresponding to activities in the network, we must first extract the activities from the raw smart card data. The method described in the following section is an extension of the implementation described in [6].

3.1 From raw transactions to journeys

The first step in preparing our data for analysis, is to derive a data set of journeys from the raw smart card data. In order to do this, we sort the raw smart card data based on the media ID and the time stamps, such that we can easily process the transactions per media ID in the order took place. We can then pass through all transactions and every time we detect a check in followed by a check out, we generate a trip, containing a departure time, departure location, arrival time and arrival location. For some modes (bus and tram) a journey may consist of several trips. After the trip-construction we merge all trips that take place within the operator specified allowed transfer time. If we end up with journeys that start and end at the same station, we remove them from the final set of journeys.

3.2 From journeys to activities and time intervals

After our first step of the process we have obtained a sequence of journeys $j_1, j_2, \ldots, j_n$ for each smart card, ordered by their departure times. For every index i such that journey i's arrival location is equal to journey $i+1$'s departure location, we create an activity starting at the arrival time of journey i, ending at the departure time of journey $i+1$ and taking place at the common location.

As we want to detect important time intervals and it is possible that our activities span multiple days, we define intervals as activities that are discretised and projected onto a modular ring. Let us first pick a number of time slots U. Throughout this paper we will work with $U = 24$, which will be interpreted as hourly time slots. All calculation involving the intervals will now be done on the modular ring $\mathbb{Z}_U$. Under the assumption that $\mathbb{Z}_U$ represents a day, the begin time of the activity is projected onto the ring, rounding the final time slot down after scaling, while the end time of the activity is rounded up after scaling. This gives us an interval $x = (x_b, x_e)$ which starts at a time slot $x_b \in 0, 1, \ldots, U-1$ and ends at time slot $x_e \in 0, 1, \ldots, U-1$. As a result, a time slot can be an "overnight" time slot in case $x_b > x_e$. For such time slots, it is not correct to take the difference $x_b - x_e$ to calculate the duration of the time slot, as time moves forward. To overcome this fact we define the duration x_d of an interval x as follows:

$$x_d = \begin{cases} x_e - x_b & \text{if } x_e \geq x_b \\ x_e + U - x_b & \text{otherwise} \end{cases}$$

4 Extracting Frequent Time Intervals by Clustering

As the number of different intervals observed at each station is likely to be too large for regular interpretation, we will apply a clustering algorithm in order to find a small number of intervals that give a compact description of the types of intervals observed at the station. At the heart of all methods for clustering is the distance measure. As the unsimilarity of two time intervals may depend of the context of the activities, we introduce a parameterised distance measure. As an example of such differences, consider that activities at an office will likely have high similarity in the starting time of the activity, while shopping or entertainment activities are more likely to have similarity in the duration.

After processing the raw smart card data, we end up with a set of stations S and a multiset I_s of observed intervals at a each station $s \in S$. We will apply[1] the k-means++ algorithm [2] on each multiset I_s. Since there are many stations in the network, we also propose a method to aggregate the cluster outputs to a full network level. The reason we do not apply the clustering algorithm on the union of all I_s multisets is that we are also interested in time intervals that occur frequently at a station that does not serve a large part of the total demand.

[1] We applied the implementation offered by the Apache Math Commons library, version 3.0. It is available at `http://commons.apache.org`

Finally, as the results of the clustering algorithm may vary with the random initial configuration, the parameterisation of the distance measure and the choice for k, we run our clustering and aggregation method multiple times in order to get a feeling for the robustness of the cluster centroids.

4.1 The parameterised distance measure

In order to assign different penalties to the distance between start time, duration and end time of the activities, we introduce a vector $\theta = (\theta_1, \theta_2, \theta_3)$. Our proposed distance measure is parameterised by this vector and can be calculated as follows:

$$d_\theta(x,y) = \begin{cases} \theta_1(x_d - y_d)^2 & \text{if } x_b = y_b \vee x_e = y_e \\ \theta_2(x_b - y_b)^2 & \text{if } x_d = y_d \\ \theta_3(|x_b - y_b| + |x_d - y_d|)^2 & \text{otherwise} \end{cases}$$

The reason to consider the three cases separately, is that we can easily control both the penalty if either the duration, the start or the end time is equal (by changing θ_1 and θ_2), and when the interval is different in every aspect (by changing θ_3).

In addition to the distance measure, we also need a way to calculate the centroid of a cluster. Since we work with $\mathbb{Z}_U$, the set of all intervals is given by $\mathbb{Z}_U^2$. In case of $U = 24$ this gives us 576 intervals. As a result, the best cluster center within a cluster of size n can be brute forced in $576 \cdot n$ calls to the distance measure.

4.2 Calculating the relevant cluster centroids and their robustness

In order to aggregate the clustering output of the individual stations, we decided to work with a threshold-based rule. This rule works as follows: given a threshold t, an interval (x_b, x_e) is relevant if there exists a station $s \in S$ such that (x_b, x_e) occurs as a centroid in the cluster output of the multiset I_s and that cluster contains more than $t|I_s|$ elements of I_s. The set of all intervals that adhere to this criterion can be calculated using Algorithm 1. We also keep track of a weight map w, which registers the fraction of the population covered by the interval in the cases where it exceeds the threshold.

Input : Distance measure parameters θ, the number of clusters k, a random seed σ, a threshold t
Output: A set of relevant intervals R, a weight map w

```
Method runExperiment(I, θ, k, σ, t) :
    foreach station s ∈ S do
        R ← ∅;
        C_1, ..., C_k ← k-means++ applied on I_s with distance measure d_θ and random seed σ;
        for i ∈ 1, ..., k do
            if |C_i|/|I_s| ≥ t then
                x ← centroid of C_i;
                R ← R ∪ {x};
                w(x) ← w(x) + |C_i|/(|I_s||S|);
            end
        end
    end
    return (R,w)
end
```

Algorithm 1: Iterative loop used to calculate relevant time intervals in the network

Since the output of the clustering algorithm, and therefore the output of the *runExperiment* method can vary for different configurations of the parameters, we decided to apply it multiple times, keeping track of how often each interval shows up. Since the number of intervals in a single result set can still be quite large, we truncate the result set to the m highest scoring intervals according to the weight map w. We then count the number of times an interval is in a truncated result set and report this as the fraction of the total number of experiments as the "*robustness fraction*". The loop we use to calculate the *robustness fractions* is presented in Algorithm 2.

Input : A set $\mathcal{C}$ of configuration parameters for runExperiment, a cutoff number m
Output: A table with for each (x_b, x_e) interval the robustness fraction

```
Method calcRobustness(C) :
    r ← new table of dimension U × U filled with 0-values;
    foreach (θ, k, σ, t) ∈ C do
        (J, w) ←runExperiment(θ, k, σ, t);
        foreach (x_b, x_e) ∈ J do
            if w((x_b, x_e)) ≥ the m highest values in {w(x) : x ∈ J} then
                r[x_b][x_e] ← r[x_b][x_e] + 100/|C|;
            end
        end
    end
    return r
end
```

Algorithm 2: Iterative loop used to calculate the robustness fraction.

5 Classification and Activity Chain Analysis

After we have applied the clustering algorithm to learn important time intervals in the network, we want to learn something about the relationship between the activities that take place during these intervals. Utilising the output of the clustering algorithm, we can propose a classification algorithm that assigns each interval to a class. This algorithm then allows us to transform the chains of activities observed in the data of the separate passengers into chains of activity classes.

5.1 Developing the classification algorithm

In [7] it was observed that there are differences in the extend to which different time intervals show up in different networks. Since our intervals are described by two time slots, it is straightforward to visualise the robustness fractions in a grid containing all possible intervals. Such a plot gives great insight in the extend to which time intervals are important. We can use the plot to propose a linear tree classifier by checking proposed rules against the robust intervals in the plot.

An important aspect to take into account during the development of the classifier is the interpretability of the chosen classes. As our focus is currently mostly exploratory, we decided to focus on classes that are easy to interpret, such as long, short, early, late and overnight.

5.2 Analysing consecutive activities

Utilising the classifier developed in the previous section we can now analyse chains of activity classes. We count all consecutive pairs of activities that are performed by the same person and are connected by a single journey. The resulting table of counts can then be interpreted as the adjacency matrix of a weighted directed graph, where the nodes represent the activity classes and the arcs represent the "followed by" relationship as observed in the data. There are many software packages available that allow us to visualise and analyse such networks. During our experiments, we have worked with Gephi [5].

Ending time (x_e) ↓ \ Starting time (x_b) →	6	7	8	9	10	11	12	13	14	15	16	17	18	19
8			-							0.11	1.00	0.99		
9				-						0.03	0.90	1.00	1.00	
10					-		0.37	0.17	0.02	0.25	0.65	0.05	0.05	0.04
11				0.87	0.02	-		0.58	0.62	0.86	0.34	0.05	0.00	0.00
12				1.00	1.00		-		0.61	0.75	0.13			
13			0.28	0.85	1.00	1.00		-	0.02	0.98	0.39			
14				0.16	0.30	1.00	1.00	0.65	-	0.28	0.87	0.09		
15				0.01	0.01	0.06	1.00	1.00	0.85	-				
16		0.44	0.37	0.04			0.62	1.00	1.00		-			
17	1.00	1.00	1.00	0.03			0.07	0.98	1.00	1.00		-		
18		1.00	1.00					0.10		0.83	0.80		-	
19											0.77	0.02		-
20												0.66		
21													0.01	

Figure 1: The table of the robustness fractions of the intervals as calculated by our clustering and aggregation algorithm.

We can count activity chains of an arbitrary length in a similar fashion. We believe that counting chains that are very long will not give a lot of insight, as passengers are not likely to perform many activities within a single day. However, smaller chain lengths, such as three or four activities, could be interesting as these patterns are likely to represent behaviour over one or two days. For this reason we added a triplet counting routine to our implementation of the processing algorithm for the adjacency matrix generation.

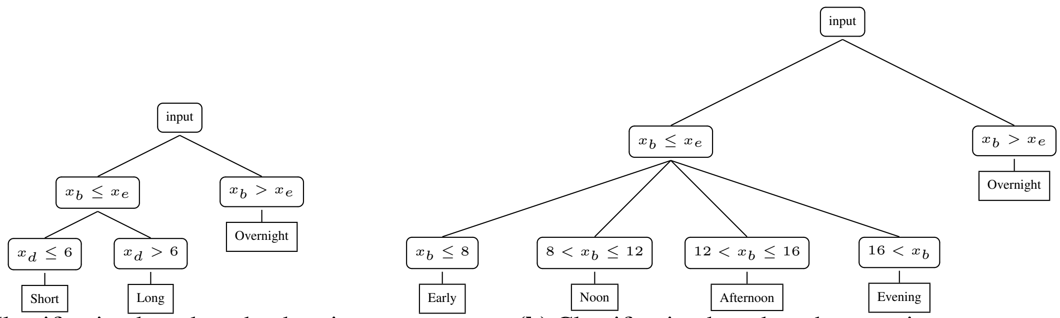
6 Experiments and Results

We have applied our clustering method on urban smart card data set from a Dutch network, containing four months of transactions. The data set contains roughly $22 \cdot 10^6$ journeys and $12 \cdot 10^6$ activities. We calculated the *robustness fractions* using the method described in Section 4.2. Our set of configurations contained all combinations of the following: for k one of $\{6, 8, 10, 20\}$, $\theta \in \{1, 2, 4\}^3$ with the constraint that $\theta_3 \geq \theta_1 \wedge \theta_3 \geq \theta_2$, one of two random seeds, $m = 40$ and $t = 0.1$. The total number of configurations is 112. The resulting table is visualised in Table 1.

Many of the highly robust intervals in Figure 1 are typically associated with commuting patterns. However, many shorter intervals that start after 9 are quite robust as well. Additionally, intervals with a duration of precisely 6 or 7 hours are very infrequent. This tells us 6 hours is a natural boundary to distinguish between short and long activities. The proposed classification tree based on the duration is presented in Figure 2a. Distinguishing between starting times appears to be more complicated. Before 9:00 short activities rarely begin, so 8:00 seems to be a good boundary for early activities. At 13:00 it seems that among the shorter activities, the intervals that are one hour longer than those starting before 13:00 become robust as well. As a result, we pick the second boundary at 12:00. Finally, after 16:00 the intervals or not very robust, so this gives us the final boundary. As it would be hard to interpret different classes of overnight activities, we introduce a single class "Overnight" for activities that have $x_b > x_e$. The resulting classification tree is presented in Figure 2b.

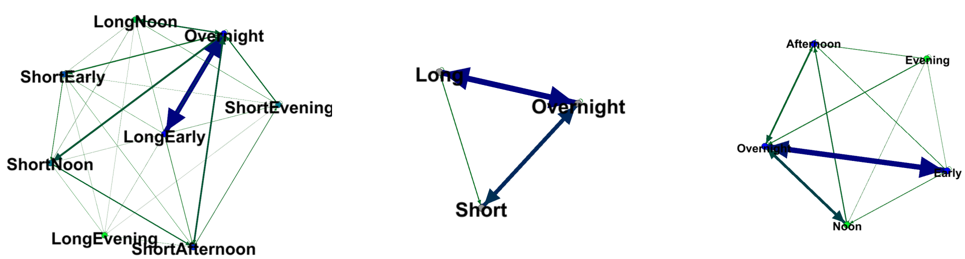
We visualised the trees for duration based classification and start time based classification separately, but they can be combined into one large classification tree. We can now apply the different classification trees to count the pairs of sequentially occurring activity classes. We visualised the pairs observed this way as a network in Figure 3

Some interesting patterns can be observed in Figure 3. First, the most prominent pairs of activities are those between an overnight interval and intervals that are early and long. These are typically time intervals associated with home-work and home-study patterns and are thus within expectation high ranking. A more interesting pattern occurs between overnight and noon activities. There is less interaction between early



(a) Classification based on the duration **(b)** Classification based on the start time

Figure 2: Classification trees for the class of an interval



(a) Combined classification (b) Duration based classification (c) Start time based classification

Figure 3: Network visualisation of the adjacency matrices based on the different classifiers

activities and noon activities than between overnight and noon activities.

Let us now consider the top ten of triplets occurring in the activity class chains. The most dominant triplets are typically associated with home - work - home like chains. The fourth and fifth triplet in the full classification describe a single activity during noon. Here we might be a bit careful in classifying the overnight activity as home: maybe some people travel to their work by car and use the public transport system during lunch time to visit a nearby location. The 10th triplet shows a pattern where two activities are started within the noon window. There is also evidence of people performing a long activity one day and a short activity the next day, and vice versa, as witnessed by triplets eight and nine in the duration based classification.

When we compare these results to the analysis of the "other" activity class considered during the analysis of Gautineau data by [7], we see that our the third triplet in the start time based classification suggest a possible peak around 12:00. However, they also found a peak around 16:00, which would be the afternoon class in our case. However, if we consider the start time classification table, only the sixth and ninth triplets represent evening activities and both are not as strong as the single noon triplet.

Full classification				Duration based classification				Start time based classification			
Overnight	LongEarly	Overnight	19%	Overnight	Long	Overnight	23%	Overnight	Early	Overnight	20%
LongEarly	Overnight	LongEarly	16%	Long	Overnight	Long	20%	Early	Overnight	Early	19%
Overnight	LongNoon	Overnight	4%	Short	Overnight	Short	10%	Overnight	Noon	Overnight	7%
Overnight	ShortNoon	Overnight	3%	Short	Short	Short	9%	Noon	Overnight	Noon	5%
ShortNoon	Overnight	ShortNoon	2%	Overnight	Short	Overnight	7%	Noon	Overnight	Early	4%
ShortAfternoon	Overnight	ShortNoon	2%	Short	Short	Overnight	6%	Afternoon	Overnight	Noon	3%
Overnight	ShortEarly	Overnight	2%	Overnight	Short	Short	6%	Early	Overnight	Noon	2%
LongNoon	Overnight	LongNoon	2%	Short	Overnight	Long	6%	Afternoon	Overnight	Early	2%
ShortAfternoon	ShortAfternoon	Overnight	2%	Long	Overnight	Short	4%	Overnight	Afternoon	Overnight	2%
Overnight	ShortNoon	ShortNoon	2%	Overnight	Long	Short	2%	Overnight	Early	Noon	2%

Table 1: The most frequent triplets for each classification method and the percentage with which they occurs among all triplets detected

7 Conclusions and Future Work

We have developed an approach to cluster temporal intervals derived from activity data at a station level using a parameterisable distance measure and to aggregate the results, such that we obtain the most interesting time intervals in the data. We repeat this process to obtain robustness fractions. Based on the robustness fractions, we constructed a linear tree classifier. The classifiers allow us to find the most frequent pairs and triplets of activity types observed in individual activity chains. While the typical intervals associated with home and work activities are dominant, we are able to identify shorter activities as well and provide some insight on their relation to other activities within the activity chains of individual passengers.

Aside from further refinements of our methods, such as reducing the number of parameters to set and varying distance measures, there are two main topics for future research. First, we would like to use either the clustering output at the station level or the complete distribution of intervals observed at the stations to identify similar classes of stations. If we are able to reduce our stations to a small number of important classes, we could include some spatial aspects in the analysis of the activity chains. We would also like to include use our findings in order to generate input for an activity based agent simulation, such as MATSim has implemented. It would be interesting to see how accurate the observed traffic counts in the smart card data can be replicated using only a small set of typical activities.

References

[1] B. Agard, C. Morency, and M. Trépanier. Mining public transport user behaviour from smart card data. In *12th IFAC Symposium on Information Control Problems in Manufacturing-INCOM*, pages 17–19, 2006.

[2] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[3] K.W. Axhausen and T. Gärling. Activity-based approaches to travel analysis: conceptual frameworks, models, and research problems. *Transport Reviews*, 12(4):323–341, 1992.

[4] M. Balmer, K. Meister, M. Rieser, K. Nagel, and K.W. Axhausen. Agent-based simulation of travel demand: Structure and computational performance of matsim-t. Technical report, ETH, Eidgenössische Technische Hochschule Zürich, IVT Institut für Verkehrsplanung und Transportsysteme, 2008.

[5] M. Bastian, S. Heymann, and M. Jacomy. Gephi: An open source software for exploring and manipulating networks. In *Proceedings of the Third International Conference on Weblogs and Social Media*, 2009.

[6] P.C. Bouman, M. Lovric, T. Li, E. van der Hurk, L.G. Kroon, and P.H.M. Vervest. Recognizing demand patterns from smart card data for agent-based micro-simulation of public transport. In *Proceedings of the Seventh Workshop on Agents In Traffic and Transportation*, 2012.

[7] F. Devillaine, M. Munizaga, and M. Trépanier. Detection of activities of public transport users by analyzing smart card data. *Transportation Research Record: Journal of the Transportation Research Board*, 2276(1):48–55, 2012.

[8] N. Eagle and A. S. Pentland. Eigenbehaviors: Identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(7):1057–1066, 2009.

[9] C. Morency, M. Trépanier, and B. Agard. Measuring transit use variability with smart-card data. *Transport Policy*, 14(3):193–203, 2007.

[10] M.-P. Pelletier, M. Trépanier, and C. Morency. Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies*, 19(4):557–568, 2011.