Urban Traffic Modelling and Prediction Using Large Scale Taxi GPS Traces

Pablo Samuel Castro[1], Daqing Zhang[1], and Shijian Li[2]

[1] Institut Telecom SudParis, 9, rue Charles Fourier; 91011 Evry Cedex, France
[2] Zhejiang University, Hangzhou, 310027, P.R. China

Abstract. Monitoring, predicting and understanding traffic conditions in a city is an important problem for city planning and environmental monitoring. GPS-equipped taxis can be viewed as pervasive sensors and the large-scale digital traces produced allow us to have a unique view of the underlying dynamics of a city's road network. In this paper, we propose a method to construct a model of traffic density based on large scale taxi traces. This model can be used to predict future traffic conditions and estimate the effect of emissions on the city's air quality. We argue that considering traffic density on its own is insufficient for a deep understanding of the underlying traffic dynamics, and hence propose a novel method for automatically determining the capacity of each road segment. We evaluate our methods on a large scale database of taxi GPS logs and demonstrate their outstanding performance.

1 Introduction

With soaring birth rates and increasing migration into urban areas, road networks are often used well over their intended capacity. This has led many cities to enforce certain measures such as restricting which vehicles can be used based on their licence plate number. There is thus an urgent need to understand, and ideally predict, the traffic dynamics in a city, not only for reducing traffic congestion, but to also address environmental, economic, and societal needs in support of a sustainable future. In the past, this has been done by installing traffic sensors in different areas of the city and extrapolating these readings throughout the city [1,2,3]. This information is usually only informative for the highways where these sensors are installed, and shed little light on the traffic flow in the rest of the city.

The last few years have seen a dramatic increase in the presence of GPS devices in vehicles for localization and/or navigation purposes. Aside from providing the user with location-based services, these devices have the potential of providing researchers with an unprecedented window into the dynamics and mobility of a city's road network. GPS-equipped taxis, in particular, have proven to be an extremely useful data source for uncovering the underlying traffic behaviour of a city. GPS-enabled taxis can be considered as ubiquitous mobile sensors constantly probing a city's rhythm and pulse. So far they have been

J. Kay et al. (Eds.): Pervasive 2012, LNCS 7319, pp. 57–72, 2012.

used for automatic map construction [4], detecting hot spots [5,6], urban computing [7,8,9,10], and characterizing passenger finding strategies [11,12,13,14,15], amongst others.

The aspect of the traffic dynamics we concern ourselves with in this paper is traffic flow and congestion, which is crucial for city planning and environmental monitoring. We measure the *flow* of traffic via the densities (*i.e.* the number of vehicles present) at each road, and consider a road segment *congested* when it has become "saturated" as a result of a high density. A deep understanding of the flow and congestion levels can be useful for improved road network design and maintenance, reducing travel times, as well as for analyzing certain side-effects of vehicle use, such as estimating pollution levels in a city [16].

There are two important problems that we address in order to approach a better understanding of traffic flow and congestion. The first is accurate prediction of future traffic conditions. We address this problem by constructing a *model* of the flow of traffic in the city, based on historical observations. The second is a principled mechanism for determining when a road segment is "over-saturated". We address this problem by examining the distribution of GPS readings in a graph plotting density versus speed.

The three main contributions of this paper can be summarized as follows.

- We define a method for constructing a *model* of the traffic flow in a city, based on the historical taxi GPS logs. This model can be used to predict future traffic conditions based on the current state.
- We argue that predicting density levels is not sufficient to obtain a thorough understanding of the traffic dynamics in a city, and present a method for automatically determining the *capacity* of roads. Coupling these computed capacities with the "raw" densities grants us a deeper understanding of the dynamics and congestion levels of a city's road network.
- We evaluate our methods on a large scale database of 5000 taxis logging their GPS information every minute (resulting in over 300 million GPS entries). Our methods are fast and simple to construct, yet they still enjoy a remarkably high degree of accuracy (less than 3% of the error incurred by the baseline considered).

The paper is organized as follows. In section 2 we discuss related work. In section 3 we describe our data set and how the data was prepared for the ensuing work. We present our predictive method along with empirical results in section 4 and our method for computing capacities in section 5. We conclude our work and discuss future avenues of research in section 6.

2 Related Work

In this section we review some of the existing work most closely related to ours. There are a number of works that propose methods for *monitoring* traffic conditions, but not necessarily modelling or predicting traffic conditions. Wen et al. [17] used GPS-equipped taxis to analyze traffic congestion changes around

the Olympic games in Beijing; note that this is an ex post facto analysis of traffic conditions. Schäfer et al. [18] used GPS-enabled vehicles to obtain real-time traffic information in a number of European cities. By considering congested roads as those where the velocity is below 10km/hr, the authors demonstrate a visualization of traffic conditions around the city can be used to detect congested and blocked road segments.

Closely related to estimating traffic conditions are obtaining accurate estimates of the travel time between two points in a city. Blandin et al. [19] use kernel methods [20] to obtain a non-linear estimate of travel times on "arterial" roads; the performance of this estimate is then improved through kernel regression. Yuan and Zheng [21] propose constructing a graph whose nodes are *landmarks*. Landmarks are defined as road segments frequently traversed by taxis. They propose a method to adaptively split a day into different time segments, based on the variance and entropy of the travel time between landmarks. This results in an estimate of the distributions of the travel times between landmarks.

The research most relevant to this paper are those which attempt to model and/or predict traffic conditions. Gühnemann et al. [16] use GPS data to construct travel time and speed estimates for each road segment, which are in turn used to estimate emission levels in different parts of the city. Their estimates are obtained by simply averaging over the most recent GPS entries; this is closely related to the historical means baseline we compare our algorithm against. Šingliar and Hauskrecht [1] studied two models for traffic density estimation: conditional autoregressive models and mixture of Gaussian trees. This work was designed to work with a set of traffic sensors placed around the city, and not with GPS-equipped vehicles. The authors assume the Markov property for traffic flows: the state of a road segment in the immediate future is dependent *only* on the state of its immediate neighbours. We adopt a similar assumption in our construction of a model. Lippi et al. [3] use Markov logic networks to perform relational learning for traffic forecasting on multiple simultaneous locations, and at different steps in the future. This work is also designed for dealing with a set of traffic sensors around the city. Su and Yu [2] used a Genetic Algorithm to select the parameters of a SVM, trained to predict short-term traffic conditions. Their method is meant to work with either traffic sensors or GPS-equipped vehicles. However, their empirical evaluation is quite limited and falls short of fully convincing the reader of their method's practicality. Herring et al. [22] use Coupled Hidden Markov Models [23] for estimating traffic conditions on arterial roads. They propose a sophisticated model based on traffic theory which yields good results. Nevertheless, we argue that this type of sophistication is, in a sense, "overkill". We capitalize on the coarse regularity of traffic flow during the week to construct a model which yields very good results, without having to resort to more sophisticated, and computationally expensive, methods. One of the main motivations driving our work is the application of these results in a real-time setting, where computationally expensive proposals are unsuitable. Yuan et al. [24] used both historical patterns and real-time traffic to estimate traffic conditions. However, the predictions they provide are between a set of "landmarks"

which is smaller than the size of the road network. Although suitable for many applications (such as optimal route planning), the coarseness of their predictions make them less suited for a detailed understanding of a city's traffic dynamics.

3 Data Preparation

In this section we present the taxi GPS data set that will be used for this paper. The raw data must be prepared in order for it to be suitable for the work discussed in the sequel and we describe this process below.

3.1 Data Set Description

For this work, we make use of a large data set obtained from around 5000 taxis in Hangzhou, China, over a period of a month (February, 2010), at a rate of approximately once per minute (for each taxi), resulting in over 300 million GPS entries. Table 1 lists the fields for each GPS entry, along with a sample entry.

Table 1. Fields for a GPS entry with a sample

Taxi ID	Longitude	Latitude	Speed (km/hr)	Bearing	Occupied flag	Year	Month	Day	Hour	Minute	Second
10429	120.214134	30.212818	70.38	240.00	1	2010	2	7	17	40	46

3.2 Mapping to a Digital Map

The "terrain" of a city is a continuous two-dimensional area (*i.e.* a subset of $\mathbb{R}^2$), which is difficult to work with. It is more practical to decompose the city into separate (usually disjoint) areas, and work with this decomposition. The way the city is decomposed is crucial to the significance of the results obtained, as the methodologies presented later are defined with respect to the way the city is decomposed. A simple and popular decomposition is to split the city's area into a matrix of disjoint grid cells; however, this has the disadvantage that one grid cell can contain more than one road segment, and it can also split road segments in unintuitive positions.

For our purposes, it is much more useful to use a digital map of the city, as this exactly represents the road network navigated by the various taxis, and it allows us to provide traffic models and/or predictions for each road.

Definition 1. *A* **digital map** *is a graph* (V, E) *where* V *is a set of vertices and* E *is a set of edges. Each edge* $e \in E$ *has the following fields: two endpoint vertices* e_{v_1} *and* e_{v_2}*; length* e_{length} *and bearing* e_{brng}*.*[1]

[1] Note that the bearing is the direction from e_{v_1} to e_{v_2}; simply subtract this bearing from 360 to obtain the bearing from e_{v_2} to e_{v_1}.

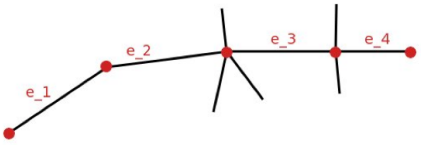

Fig. 1. A street split into various edges

A street can consist of multiple edges, as shown in Figure 1. Note that it is split into segments wherever there is an intersection and/or a change in bearing.

In many cases, a digital map of the city is not readily available. Given the long time span and large number of taxis available in our data set, the city's road network becomes apparent by simply plotting all the GPS entries. In the left panel of Figure 2 we display a sample of the plot obtained from 20 taxis over one month, around the downtown area of Hangzhou. We only plot the pickup and dropoff points, as we found that including full trajectories clutters the plot. A simple (but arduous) option is to "draw" the edges and construct a digital map manually. We split roads into segments at every interesection, as well as anytime the bearing of the road changes considerably. In the right panel of Figure 2 we display the resulting digital map. It consists of 2003 edges and 1585 vertices.

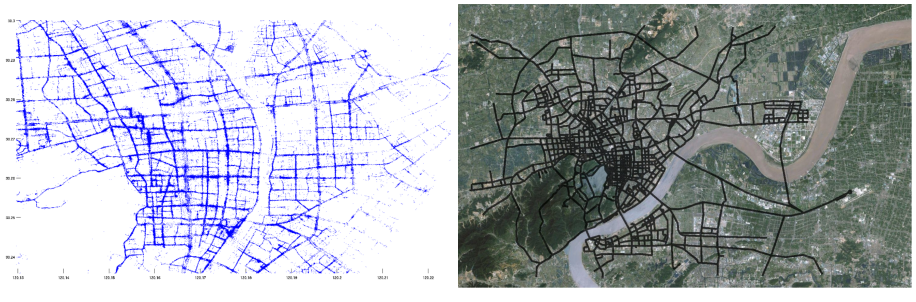

Fig. 2. Left: Traces from 20 taxis over one month. Right: Digital road network drawn over raw traces.

Given a digital map (V, E), one can map each GPS entry to a point on one of the edges from E. Simply mapping to the closest edge may not produce good results, and errors will frequently occur, as shown in the left panel of Figure 3. There are number of approaches to map-matching that take contextual information into consideration such as edit distance [25] and Fréchet distance [26,27]. These approaches map trajectories *globally*, that is, only complete trajectories are considered. This renders these approaches unsuitable for long trajectories, as the computational expense becomes too large. We perform our map-matching on a *local* basis, using contextual information such as distance and orientation (see the right panel of Figure 3). In this example, we are able to avoid mapping (erroneously) to road segment 2 on the left hand side by comparing the orientation of this road segment to the orientation indicated by the GPS logs.

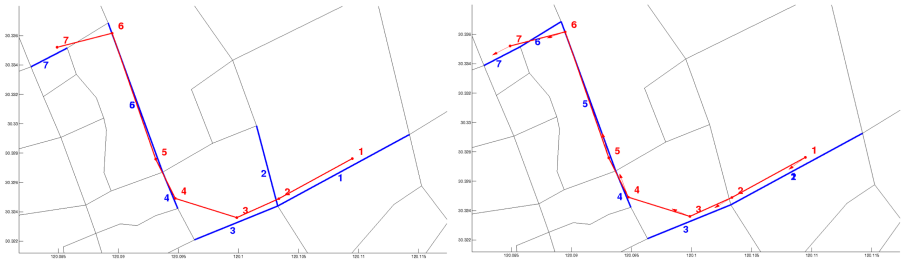

Fig. 3. Mapping trajectories to a digital map, the original trajectory is shown in red, the edge each entry is mapped to is shown in mauve. Left: Mapping each entry to the closest edge; Right: Using bearing information of each GPS entry to improve edge matching.

A GPS entry that has been mapped onto a digital map will result in a new entry, identical except for the latitude and longitude which may be different (so that the point "sits on" the digital map). This new entry is augmented with the fields: *edge* and *orien*. The field *edge* represents what edge $e \in E$ it was mapped to, and *orien* represents what orientation it is following (1 if going from e_{v_1} to e_{v_2} and 2 if going in the other direction).

For the rest of the paper, we will always consider an edge e with an orientation o. For clarity, we will refer to this pair as (e, o) with the following attributes:

$$(e,o)_{v_1} = \begin{cases} e_{v_1} & \text{if } o = 1 \\ e_{v_2} & \text{otherwise} \end{cases}$$

$$(e,o)_{v_2} = \begin{cases} e_{v_2} & \text{if } o = 1 \\ e_{v_1} & \text{otherwise} \end{cases}$$

4 Predictive Model

Now that we have described the data set and the digital map decomposition, we can proceed to describe our predictive model. The main idea is to use the navigational history to model the flow of traffic at different times. We are assuming there is a form of regularity present in the way traffic flows at different times. Although this may seem like a strong assumption, it is justified by the empirical results presented below. Before we describe the predictive model we will describe the way we collect our data and gather the necessary statistics. We will make use of the successors of an edge-orientation pair, defined below.

Definition 2. *The* **successors** *of an edge e with orientation o is defined as the set of* consistent *adjacent edge-orientation pairs:*

$$succ((e,o)) = \{(e',o') | (e,o)_{v_2} = (e',o')_{v_1}\}$$

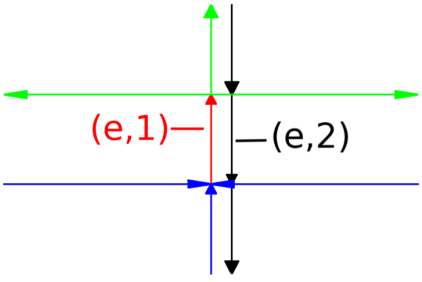

Fig. 4. Successors (in green) and predecessors (in blue) of edge-orientation pair (e, 1) (in red). Note that the vertical arrows represent the two orientations of the same road.

For convenience, we can define the predecessors of a pair (e, o) as

$$pred((e, o)) = \{(e', o') | (e, o) \in succ((e', o'))\}$$

See Figure 4 for clarification.

Since the sampling rate of the GPS devices is once per minute, we will use an initial granularity of one minute, yielding 1440 minutes for each day. For every edge-orientation pair (e, o) and minute m, we maintain the function $moveCounts((e, o), m)((e', o'))$, which is simply the number of times a transition was made from (e, o) to (e', o') in minute m. Note that this function is well-defined even when $(e, o) = (e', o')$, as this simply counts the number of times a taxi *did not* transition to another edge in minute m.

With the above function, we can compute a probabilistic transition matrix for every minute m, as follows:

$$P_m((e, o), (e', o')) = \frac{moveCounts((e, o), m)((e', o'))}{\sum_{(e'', o'')} moveCounts((e, o), m)((e'', o''))}$$

The above formula is simply computing the frequency of the available edge-orientation transitions. Note that, as in Singliar and Hauskrecht [1], we are assuming the Markov property: the state of traffic of an edge-orientation pair in the immediate future is dependent *solely* on the state of its immediate neighbours.

In practice, there is significant variability in the transition distributions between successive minutes, so we propose re-defining this function with a granularity of 15-minute segments, yielding 96 segments per day. Given a 15-minute quarter q, we define the probabilistic transition matrix as follows:

$$P_q((e, o), (e', o')) = \frac{\sum_{m \in q} P_m((e, o), (e', o'))}{15}$$

Henceforth we will only make use of the transition matrix defined over 15-minute quarters, so there will be no risk of ambiguity between P_m and P_q. Note that the size of each matrix P_q is $|E| * 2 \times |E| * 2$ (for our data set this results in a matrix of size 4406×4406).

The purpose of this transition function is to be able to model the flow of traffic in the city. In this paper, we consider traffic as the *density* of taxis. Since we have a large number of taxis continuously driving around the city, we believe they are a good indication of the true vehicular density. We define $D_q((e,o))$ as the number of taxis on (e,o) in 15-minute quarter q. Clearly, the higher the density in a road segment, the higher the level of traffic. Each vector D_q has size $|E| \times 2$.

We can now present our method for predicting the traffic density in the next quarter $q+1$. The main idea is that we predict the next-step densities by *spreading* the current density according to the transition matrix. For instance, the proportion of the current density at (e,o), $D_q((e,o))$, that will "flow" to (e',o') is given by $P_q((e,o),(e',o'))$; similarly the proportion of the current density at (e,o) that will not move is given by $P_q((e,o),(e,o))$. Thus, the predicted density of (e',o') in quarter $q+1$ is computed as follows:

$$\hat{D}_{q+1}((e',o')) = \sum_{(e,o)} D_q((e,o)) P_q((e,o),(e',o')) \tag{1}$$

Note that $P_q((e,o),(e',o'))$ will only be positive for $(e,o) \in pred((e',o'))$.

We can compute Equation 1 for all edge-orientation pairs by expressing it as a matrix operation:

$$\hat{D}_{q+1} = P'_q \cdot D_q \tag{2}$$

where P'_q is the transpose of P_q. We may also desire to obtain a prediction for further in the future, which can be easily done by successive applications of the transition matrices. The n-step transition matrix from q is defined as follows.[2]

$$P_q^{(n)} = P_q \cdot P_{q+1} \cdots P_{q+n-1}$$

Obtaining a prediction for n steps in the future is then straightforward:

$$\hat{D}_{q+n} = P_q^{(n)'} \cdot D_q \tag{3}$$

4.1 Empirical Evaluation

In this section we aim to measure the accuracy of our prediction method. Given that traffic congestions and overall vehicular volume is greater during weekdays, we focus on weekdays. The same procedure can of course be applied to weekends; however, a model learned from working and non-working days may not be as accurate, as the traffic patterns differ significantly between the two types of days. We split our month-long data into four work-weeks, yielding 480 15-minute quarters (assuming a five day work-week). We use one of these weeks as the training set, w_I , and another as the testing set, w_T . For a given edge-orientation pair (e,o) and week w, $D_w((e,o))$ is a vector of length 480 containing the densities at (e,o) for each quarter in week w.

[2] Note that this must be done module 96, *i.e.* $95 + 2$ should not yield 97, but 1.

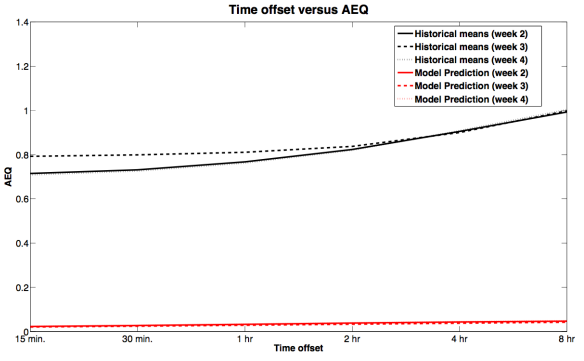

Fig. 5. Time offset versus AEQ

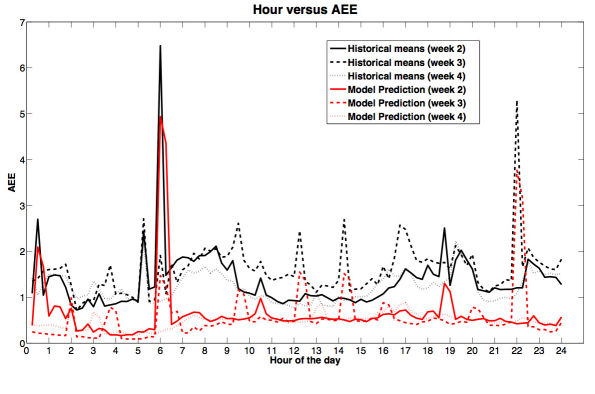

Fig. 6. Hour of the day versus AEE

We use the data from w_I to construct a model, as described in the previous section. Then, given a 15-minute quarter q and a time offset n, we compute $\hat{D}_{q+n}$ as described above. Since there is a fair amount of regularity in the traffic flow during work weeks, a simple way to "predict" traffic would be to use the historical mean of traffic density, given the current 15-minute quarter. We use this as a baseline against which we can compare our algorithm's performance.

We measure the performance of our algorithm and the baseline method using the following measures.

1. Given $\hat{D}_{q+n}$ for all values of q and for a specified offset n, we compute the average error per quarter (AEQ) as follows:

$$AEQ^n(\hat{D}) = \frac{\sum_q \|D_q - \hat{D}_q\|_2}{480} \tag{4}$$

Note that $AEQ(\hat{D})$ is a vector of size $|E| \times 2$, we average over all edges to obtain our final measure:

$$AEQ^n = \frac{\sum_{(e,o)} AEQ^n(\hat{D})((e,o))}{|E| * 2}$$

We plot the results in Figure 5 with varying values of n, where we can clearly see the advantage of using our proposed prediction method over standard means. Indeed, our method incurs less than 3% of the error incurred by the baseline model. From this figure we can also conclude that although there are some regularities in traffic flow from week to week, there is still enough variability to incur a significant error.

2. Given $\hat{D}_{q+n}$ for all values of q and for a specified offset n, we compute the average error per edge (AEE) as follows:

$$AEE^q(\hat{D}) = \frac{\sum_{(e,o)} \|D_q(e) - \hat{D}_q(e)\|_2}{|E| * 2} \quad (5)$$

Because q ranges over a full work week (480 possible values), we average over the 5 working days (resulting in 96 possible values). We plot the results when using an offset of $n = 1$ (i.e. 15 minutes) in Figure 6 for the different weeks. In this graph we can once again see the improved performance of our method, as well as the variability from week to week. We can also observe the same general "shape" present in all the weeks: there is greater error (due to greater complexity resulting from a higher volume of vehicles) during working hours (roughly from 9am to 7pm).

5 Determining Road Capacities

In the last section we demonstrated that we can predict the density at different road segments with sufficient accuracy. However, in the absence of additional information about the road segments, these density predictions may not be very useful, as different road segments get congested with different density levels, depending on their length and width. In this section we propose a novel method for determining the *capacity* of the different road segments, based on the historical density and speed readings.

For each edge-orientation pair (e, o) and minute m we compute $vel((e, o), m)$, defined as the average velocity of all taxis passing by (e, o) in minute m. In the computation of this average velocity we apply some simple filtering schemes, such as ignoring the velocity of taxis that are parked: parked taxis are not navigating the network so are not a proper indication of the average speed through (e, o), and will bring down the computed average velocity.

We can use average velocity as an indication of when a road is congested. Schäfer et al. [18] define a congested road as one where the average velocity is below 10km/hr. In this paper we say a road segment is congested if the velocity of the majority of taxis is below 20km/hr, in a manner that will be made more

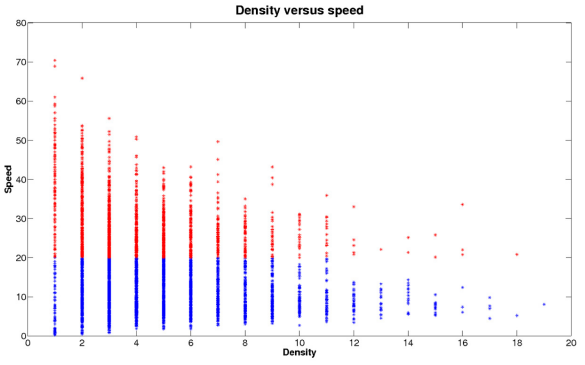

Fig. 7. Density versus speed for one edge-orientation pair

precise below. We choose 20km/hr as our cutoff as this is already quite low, even for residential areas. In Figure 7 we plot density versus average speed for a particular edge-orientation pair; that is, every point in the figure corresponds to one of the 7200 minutes in a week. We colour the points above our 20km/hr limit in red, and those below in blue. We will refer to the red points as the *high* points, and the blue points as the *low* points. As one would expect, the velocity tends to go down as the density goes up. It is inevitable that even with low densities we will have low speed readings, and simply using averages may under estimate the road's true capacity. We compute, for each density level d, the ratio of high points, $high_d$ to low points, low_d:

$$ratio(d) = \frac{|high_d|}{|low_d|}$$

We define the capacity of an edge-orientation pair (e, o), as the density level d, with sufficient data points, whose ratio unambiguously drops below 0.4 (*i.e.* there are at least 250% more low points than high points). A density level d has sufficient data points if $|high_d| + |low_d| > 500$. We say a ratio unambiguously drops below 0.4 at d if $ratio(d-1) > 0.4$ and $ratio(d+1) < 0.4$. The unambiguous drop criterion is meant to exclude outlier "spikes" that may anomalously drop below 0.4 for a single density level. We denote this capacity as $cap((e, o))$. For a pair (e, o), if the ratio never drops below 0.4 or it only drops below 0.4 when there are insufficient points, we say (e, o) does not have a capacity. In Figure 8 we plot density versus ratio as well as the number of points per density for the same edge-orientation pair from Figure 7. The red line in the graph indicates the ratio threshold (0.4) while the green line indicates the capacity determined for this edge-orientation pair.

Unfortunately, we do not have access to "ground-truth" capacities for each road segment, so it is difficult to quantitatively measure the accuracy of our capacity computation method. Nevertheless, by examining weekly traffic patterns for different road segments, we can qualitatively verify that our computed capacities are reasonable. In Figure 9 we display the density over the course of a

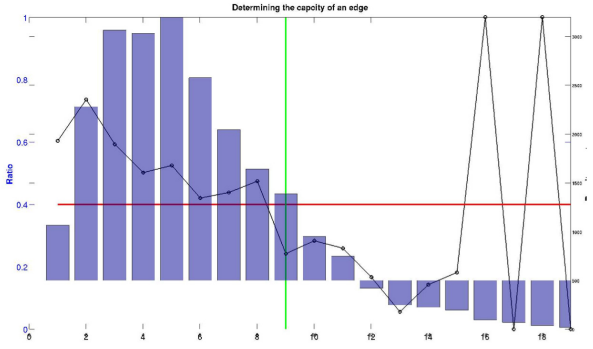

Fig. 8. Density versus ratio (line with left y-axis) and density counts (bar graph with right y-axis) for one edge-orientation pair. The red line is the ratio threshold while the green line is the determined capacity.

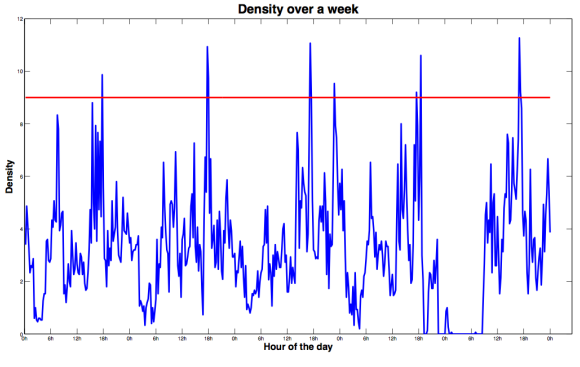

Fig. 9. Density for one edge-orientation over the course of a week, with capacity drawn in red

week for the same edge-orientation pair as in the previous two figures. We can see that this particular edge-orientation pair has density peaks at roughly 6pm every day. It is also intuitively clear that in order to detect these peaks, one would have to set the capacity level somewhere between 8 and 11. Our method sets it at 9, which is within range; however, this value would vary depending on the chosen speed limit (in our case, 20 km/hr) and the ratio threshold (in our case, 0.4). This qualitative verification was performed on a number of different road segments with similar results.

In Figure 10 we display a snapshot of the downtown area during rush hour (10am). On the top panel we display the raw densities at each road segment, where the colours cover the range of densities at the current 15-minute quarter. On the bottom panel we display the density at each road segment divided by its capacity; in other words, the proportion of the capacity of the road segment that has been filled (it may be larger than one if it is filled over its capacity). Because of the small length of the visible road segments, the visualization on

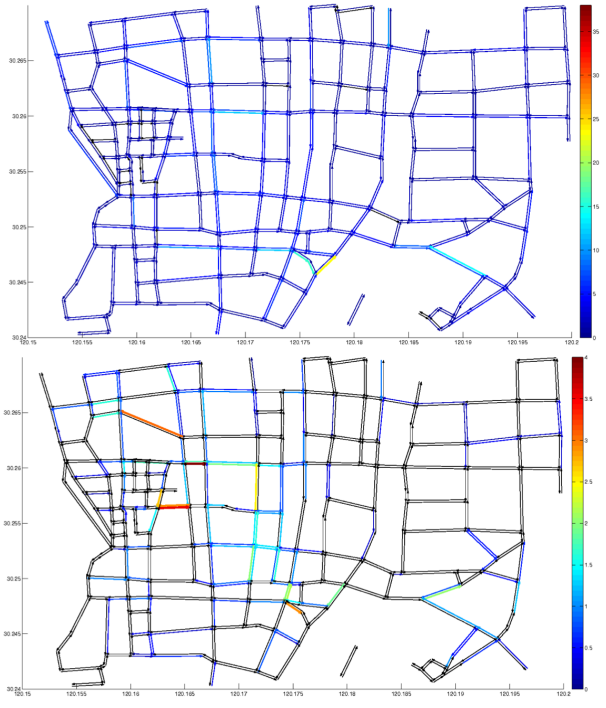

Fig. 10. Density visualization in a part of the downtown at 10am. Top: Absolute densities; Bottom: density divided by road capacity.

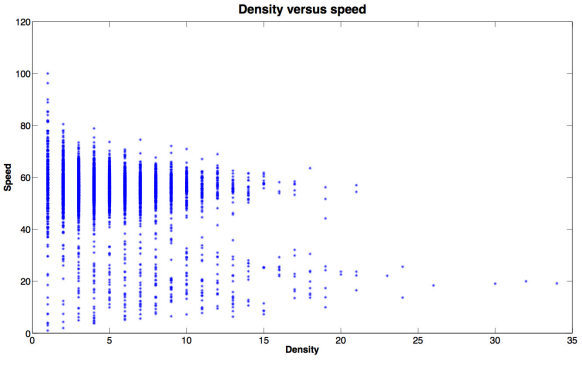

Fig. 11. Density versus speed for a bridge

top conveys very little information about the congestion levels at each of these road segments: relative to larger segments, the absolute density at each of these segments is quite small. On the other hand, when considered with respect to their capacity, we obtain a visualization that is more informative, as on the bottom. If one is interested in monitoring traffic congestion levels in a city, simply observing the absolute densities, as on top, is clearly not sufficient.

6 Conclusion

In this paper we have presented a novel method for modelling the flow of traffic in a large city by means of large scale taxi GPS trajectories. Our method avoids the computational complexity of related works [1,22] while still enjoying remarkable accuracy. The constructed model applies to all visited road segments within a city's road network, which is a finer granularity than solely using hotspots or "landmarks" [24]. Having a finer granularity grants one a more detailed view of the traffic flow; however, road segments that have been infrequently visited will suffer from inaccurate models. Nevertheless, it is precisely those road segments that have a higher visitation frequency that we are most interested in, as these are the road segments that most affect the traffic flow.

We demonstrated that our method is able to provide consistently accurate predictions, even when considering a time frame 16 hours into the future! We proved that this is not simply a consequence of the regularity of traffic densities by comparing against a simple predictor that simply uses the average historical density. It is important to point out that there are two types of regularities discussed in this paper: the first is the aforementioned regularity of traffic densities, which is insufficient for accurate predictions; the second is the regularity of localized navigation decisions. This is the regularity we referred to in the introduction, which allows us to step away from unnecessary sophistication and into a simpler, but comparably accurate, method. We believe the regularity of the second type has less variability, and *results* in the regularity of the first type, which has higher variability.

Our mechanism for automatically determining the capacity of road segments is the first of its kind, and provides a powerful tool for obtaining a deep understanding of the traffic dynamics. Besides its use for automatically determining the capacity of a road segment, the density versus speed plots (as in Figure 7) can provide additional useful information about road segments. For instance, consider Figure 11, which is the plot for one of the bridges in Hangzhou. We can see that there are two "clusters" in this plot, one centered around a speed of 60km/hr, while the other centered around 20km/hr. We believe this behaviour is a result of the presence of bus/taxi lanes in many bridges and highways, which generally have a higher speed than the other lanes. Since our data is coming from taxis, most of the data points fall in the 60km/hr cluster. This suggests that this type of plot can be useful for extracting lane information, which is a problem that has been previously considered [28,29,30].

Although our method provides a means for predicting future traffic conditions, it does not indicate the *correlation* amongst different road segments, and more specifically, the *influence* certain road segments have on other road segments [31]. This information can be extremely useful when planning the closure of certain roads for maintenance, special events, etc. By understanding this influence the road closure schedule can be planned in order to minimize the impact on the city's drivers. We are currently working on this problem, and expect the results presented in this paper to be key elements of this future work.

Acknowledgements. The authors would like to thank Lin Sun, Tiezhen Wang and Xu Qiao for their hard work in constructing the digital map of Hangzhou. The authors would also like to thank Chao Chen for his support throughout this work.

References

1. Šingliar, T., Hauskrecht, M.: Modeling Highway Traffic Volumes. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 732–739. Springer, Heidelberg (2007)
2. Su, H., Yu, S.: Hybrid GA Based Online Support Vector Machine Model for Short-Term Traffic Flow Forecasting. In: Xu, M., Zhan, Y.-W., Cao, J., Liu, Y. (eds.) APPT 2007. LNCS, vol. 4847, pp. 743–752. Springer, Heidelberg (2007)
3. Lippi, M., Bertini, M., Frasconi, P.: Collective Traffic Forecasting. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010, Part II. LNCS, vol. 6322, pp. 259–273. Springer, Heidelberg (2010)
4. Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W., Huang, Y.: Map-Matching for Low-Sampling-Rate GPS Trajectories. In: Proceedings of ACM SIGSPATIAL (2009)
5. Chang, H., Tai, Y., Hsu, J.Y.: Context-aware taxi demand hotspots prediction. International Journal of Business Intelligence and Data Mining 5(1), 3–18 (2010)
6. Liu, S., Liu, Y., Ni, L.M., Fan, J., Li, M.: Towards Mobility-based Clustering. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2010 (2010)
7. Hu, J., Cao, W., Luo, J., Yu, X.: Dynamic Modeling of Urban Population Travel Behavior based on Data Fusion of Mobile Phone Positioning Data and FCD. In: 17th International Conference on Geoinformatics (2009)
8. Liu, L., Biderman, A., Ratti, C.: Urban Mobility Landscape: Real Time Monitoring of Urban Mobility Patterns. In: Proceedings of the 11th International Conference on Computers in Urban Planning and Urban Management, CUPUM 2009 (2009)
9. Qi, G., Li, X., Li, S., Pan, G., Wang, Z.: Measuring Social Functions of City Regions from Large-scale Taxi Behaviors. In: PerCom Workshop (2011)
10. Zheng, Y., Liu, Y., Yuan, J., Xie, X.: Urban Computing with Taxicabs. In: Proceedings of the 13th ACM International Conference on Ubiquitous Computing, UBICOMP 2011 (2011)
11. Chang, H., Tai, Y., Chen, H.W., Hsu, J.Y.: iTaxi: Context-Aware Taxi Demand Hotspots Prediction Using Ontology and Data Mining Approaches. In: Proceedings of the 13th Conference on Artificial Intelligence and Applications (TAAI 2008) (2008)
12. Lee, J., Shin, I., Park, G.L.: Analysis of the passenger pick-up pattern for taxi location recommendation. In: Proceedings of the 2008 Fourth International Conference on Networked Computing and Advanced Information Management, vol. 1 (2008)
13. Liu, L., Andris, C., Ratti, C.: Uncovering cabdrivers' behavior patterns from their digital traces. Computers, Environment and Urban Systems 34, 541–548 (2010)

14. Phithakkitnukoon, S., Veloso, M., Bento, C., Biderman, A., Ratti, C.: Taxi-Aware Map: Identifying and Predicting Vacant Taxis in the City. In: de Ruyter, B., Wichert, R., Keyson, D.V., Markopoulos, P., Streitz, N., Divitini, M., Georgantas, N., Mana Gomez, A. (eds.) AmI 2010. LNCS, vol. 6439, pp. 86–95. Springer, Heidelberg (2010)
15. Li, B., Zhang, D., Sun, L., Chen, C., Li, S., Qi, G., Yang, Q.: Hunting or waiting? Discovering passenger-finding strategies from a large-scale real-world taxi dataset. In: PerCom Workshops (2011)
16. Gühnemann, A., Schäfer, R., Thiessenhusen, K.: Monitoring traffic and emissions by floating car data. Institute of transport studies Australia (2004)
17. Wen, H., Hu, Z., Guo, J., Zhu, L., Sun, J.: Operational Analysis on Beijing Road Network during the Olympic Games. Journal of Transportation Systems Engineering and Information Technology 8(6), 32–37 (2008)
18. Schäfer, R.P., Thiessenhusen, K.U., Wagner, P.: A Traffic Information System by Means of Real-Time Floating-Car Data. In: 9th World Congress on Intelligent Transport Systems (2002)
19. Blandin, S., Ghaoui, L.E., Bayen, A.: Kernel regression for travel time estimation via convex optimization. In: Proceedings of the 48th IEEE Conference on Decision and Control (2009)
20. Scholkopf, B., Smola, A.: Learning with kernels. MIT press (2002)
21. Yuan, J., Zheng, Y.: T-Drive: Driving Directions Based on Taxi Trajectories. In: ACM SIGSPATIAL GIS (2010)
22. Herring, R., Hofleitner, A., Abbeel, P., Bayen, A.: Estimating arterial traffic conditions using sparse probe data. In: Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (2010)
23. Brand, M.: Coupled hidden markov models for modeling interacting processes. Technical report, The Media Lab, Massachusetts Institute of Technology (1997)
24. Yuan, J., Zheng, Y., Xie, X., Sun, G.: Driving with Knowledge from the Physical World. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2011)
25. Yin, H., Wolfson, O.: A Weight-based map matching method in moving objects databases. In: Proceedings of the 16th International Conference on Scientific and Statistical Database Management (2004)
26. Alt, H., Efrat, A., Rote, G., Wenk, C.: Matching planar maps. Journal of Algorithms 49, 262–283 (2003)
27. Brakatsoulas, S., Pfoser, D., Salas, R., Wenk, C.: On map-matching vehicle tracking data. In: Proceedings of the 31st International Conference on Very Large Data Bases (2005)
28. Rogers, S., Langley, P., Wilson, C.: Mining GPS data to augment road models. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 1999, pp. 104–113 (1999)
29. Edelkamp, S., Schrödl, S.: Route planning and map inference with global positioning traces, pp. 128–151. Springer-Verlag New York, Inc., New York (2003)
30. Chen, Y., Krumm, J.: Probabilistic modeling of traffic lanes from GPS traces. In: Proceedings of the 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (2010)
31. Liu, W., Zheng, Y., Chawla, S., Yuan, J., Xie, X.: Discovering Spatio-Temporal Causal Interactions in Traffic Data Streams. In: Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2011)