Predicting Travel Time Reliability using Mobile Phone GPS Data

Dawn Woodard, Cornell University, Ithaca, NY*
Galina Nogin, Cornell University, Ithaca, NY†
Paul Koch, Microsoft Research, Redmond, WA
David Racz, Microsoft Corp., Palo Alto, CA
Moises Goldszmidt, Microsoft Research, Mountain View, CA‡
Eric Horvitz, Microsoft Research, Redmond, WA

March 23, 2015

Abstract

Probabilistic forecasts of travel time can be used for risk-averse routing, for reporting travel time reliability to a user, or as a component of fleet vehicle decision-support systems. Many of these uses (such as those for mapping services like Bing or Google Maps) require predictions for arbitrary routes in the road network, at arbitrary times; the highest-volume source of data for this purpose is GPS data from mobile phones. We introduce a method (“TRIP”) to predict the probability distribution of travel time on an arbitrary route in a road network at an arbitrary time, using GPS data from mobile phones or other probe vehicles. TRIP gives informed predictions for parts of the road network with little data, captures weekly cycles in congestion levels, and is computationally efficient even for very large road networks and datasets. We apply TRIP to predict travel time on the road network of the Seattle metropolitan region, based on large volumes of GPS data from Windows phones. TRIP provides improved interval predictions (forecast ranges for travel time) relative to Microsoft’s engine for travel time prediction as used in Bing Maps. It also provides deterministic predictions that are almost as accurate as Bing Maps predictions, despite using fewer explanatory variables, and differing from the observed travel times by only 10.4% on average over 36,493 test trips. To our knowledge TRIP is the first method to provide accurate predictions of travel time reliability for complete, large-scale road networks.

Keywords. Location data, traffic, travel time, forecasting, statistics.

1 Introduction

Several mapping services provide predictions of the expected travel time on an arbitrary route in a road network, in real time and using traffic, time of day, day of the week, and other information. They use these predictions to recommend a route or routes with low expected travel time. Microsoft’s mapping service (Bing Maps) predicts travel time for large-scale road networks around the world using a method called Clearflow [Microsoft Research, 2012], which employs probabilistic graphical models learned from data to...
predict flows on arbitrary road segments. The method, which has its roots in the earlier Smartphlow effort on forecasting highway flows and reliability (Horvitz et al., 2005), considers evidence about real-time traffic conditions, road classifications, topology of the road network, speed limits, time of day and day of week, and hundreds of other variables about roads, including proximity of road segments to schools, parks, and shopping malls. With Clearflow, travel time predictions made on all segments across a geographic region are used in route-planning search (Delling et al., 2015).

Beyond expected flows, it is important to consider uncertainty in travel time caused for instance by unpredictable traffic light schedules, unexpected road conditions, and differences in driver behavior. Such travel time variability (conversely, its reliability) also strongly affects the desirability of routes in the road network (Jenelius 2012; Texas Transportation Institute 2015). For fleets of delivery vehicles such as those transporting perishables, decisions including routing need to provide on-time deliveries with high probability. In the case of ambulance fleets, taking into account uncertainty in the travel time of an ambulance to potential emergency scenes leads to improved ambulance positioning decisions, and consequently increases the survival rate of cardiac arrest patients (Erkut et al., 2007). A prediction of the probability distribution of travel time can be more valuable than a deterministic prediction of travel time, by accounting not just for measured traffic congestion and other known conditions, but also for the presence of unmeasured conditions. Distribution predictions of travel time can be used for risk-averse routing, for reporting travel time reliability to a user (e.g. the travel time is predicted to be in the range 10-15 minutes), and as a component of fleet vehicle decision-support systems (Samaranayake et al., 2012; Westgate et al., 2015).

Figure 1: Anonymized GPS locations from Windows phones in the Seattle metropolitan region, aggregated into grid boxes and colored by average speed in the grid box. Orange indicates the slowest speeds, while blue indicates the fastest.

We introduce a statistical solution to predicting the distribution of travel time on an arbitrary route in the road network, at an arbitrary future time. We call the method TRIP (for travel time reliability inference and prediction). For typical road networks of interest, the number of possible routes is extremely large, and any particular route may have very few or no observed trips in the historical data. For these reasons it is infeasible to apply methods designed for prediction on a particular set of heavily traveled routes, such as Jenelius & Koutsopoulos (2013); Ramezani & Geroliminis (2012); Rahmani et al. (2015). TRIP uses
information from all the trips in the historical data to train a model for travel time on routes, learning the characteristics of individual roads and the effect of time of the week, road classification, and speed limit. We model travel time variability both at the trip level and at the level of the individual road network links included in the route. This decomposition is appropriate because some sources of variability affect the entire trip (such as driver habits, vehicle characteristics, or unexpected network-wide traffic conditions), while other sources of variability are localized (such as a delay due to a train crossing or construction). We define a network link to be a directed section of road that is not divided by an intersection, and on which the measured features of the road (road classification, speed limit, number of lanes, etc.) are constant.

TRIP captures important features of the data, including weekly cycles in congestion levels, heavy right skew of travel time distributions, and probabilistic dependence of travel times for different links within the same trip (for example, if the travel speed is high on the first link of the route, the speed is also likely to be high on the other links of the route). We capture the multimodality of travel time distributions using a mixture model where the mixture components correspond to unobserved congestion states, and model the probabilistic dependence of these congestion states across the links of the route using a Markov model. Because we model travel time for individual links, the travel time prediction can be updated en route.

We introduce a computational method for training and prediction based on maximum a posteriori estimation via Expectation Conditional Maximization (Meng & Rubin, 1993). This yields an iterative training process with closed-form update equations that can be computed using parallelization across links and trips; as a result it is computationally efficient even on large road networks and for large datasets.

TRIP uses Global Positioning System (GPS) data from vehicle trips on the road network; we obtain large volumes of such trips using anonymized mobile phone GPS data from Windows phones in the Seattle metropolitan region. We compare the accuracy of our predictions to a variety of alternative approaches, including Clearflow. The GPS location and speed measurements are illustrated in Figure 1, which shows that they contain valuable information regarding the speed of traffic on individual roads. Unlike other sources of vehicle speed information (Hofleitner et al., 2012b), vehicular GPS data does not require instrumentation on the roadway, and can achieve near-comprehensive coverage of the road network. Additionally, there is increasing evidence that traffic conditions can be estimated accurately using only vehicular GPS data (Work et al., 2010). One challenge of mobile phone GPS data is that it is often sampled at low frequency (typically 1-90 seconds between measurements). A related data source, GPS data from fleet vehicles, is also often sampled at low frequency (Rahmani et al., 2015), and TRIP can also be applied to such data.

Some existing approaches to predicting the probability distribution of travel time on a road network model exclusively link-level variability, and assume independence of travel time across the links in the route (Westgate et al., 2013; Hunter et al., 2013a). This leads to considerable underprediction of the amount of variability (Westgate et al., 2013) and Section 4. Dependence across links is incorporated by Hofleitner et al. (2012a,b), who use a mixture model for travel time on links where the mixture component represents a congestion state (as we do). They allow these congestion states to depend on the link and the time, and model dependence of their congestion states across the road network and across time using a dynamic Bayesian network. This approach is intuitive but computationally demanding (leveraging a high-dimensional particle filter in each iteration of their algorithm), so is unlikely to be efficient enough for complete road networks (they apply it to 800 links in the San Francisco arterial network). Additionally, it still underpredicts the amount of variability in travel time. Motivated by evidence in the data (Section 4), we allow the congestion state to additionally depend on the whole trip. We model dependence of this congestion state across the links of the route, instead of across all links of the network. This improves the flexibility of the model, leading to accurate variability prediction. It also facilitates computation: because our specification corresponds to a one-dimensional Markov model, computation can be done exactly and efficiently in each iteration, by using the forward-backward algorithm (cf., Russell & Norvig, 2009).
Another existing method for prediction of travel time variability (Hunter et al., 2013b) is designed for high-frequency (e.g. every second) GPS measurements. These authors estimate the number of stops of the vehicle on each link of the route, and introduce a model for the number of stops and a model for the distribution of travel time conditional on the number of stops. There are also some methods designed for emergency vehicles, which directly model the distribution of travel time on the entire trip, as a function of quantities like the route distance (Budge et al., 2010; Westgate et al., 2015). This is appropriate for emergency vehicles because the relevant data source (lights-and-sirens emergency vehicle trips) is too low-volume to effectively model the distribution of travel times for individual links; for non-emergency data it does not work as well as our approach, as we show in Section 4.

The scale of the road network for which we do prediction of travel time distributions (221,980 links) is an order of magnitude larger than existing work for non-emergency vehicles. Hunter et al. (2013b) consider a network of over half a million links, but then remove the links with too few observations; presumably this limits the predictions to routes that do not include any of the deleted links.

In Section 2 we describe our statistical model and in Section 3 we present methods for training and prediction with the model. Section 4 gives the Seattle case study, including providing support for our modeling choices and reporting our prediction results. We draw conclusions and discuss extensions in Section 5.

2 Modeling

TRIP uses GPS measurements recorded periodically during vehicle trips. Each GPS observation consists of location, speed, and heading measurements, along with a time stamp. For mobile phones, a GPS measurement is recorded whenever phone applications access GPS information, not all of which are mapping or routing applications. For this reason, the frequency of GPS measurements vary, and the phone is not necessarily in a moving vehicle at the time when the measurements are taken. However, motorized vehicle trips can be isolated using a set of heuristics; for the Seattle case study, we identify sequences of at least 3 GPS measurements from the same device that satisfy requirements such as: (a) starting and ending with speed measurements of at least 3 m/s, (b) having average speed of at least 5.5 m/s and maximum speed of at least 9 m/s, and (c) covering distance at least 1 km (as measured by the sum of the great circle distances between pairs of sequential measurements). The resulting GPS sequences appear to consist almost exclusively of motorized vehicular trips that follow paths in the road network. The requirements regarding the average and maximum speeds, for example, eliminate most trips from non-motorized travel such as biking or walking. We define each trip to start with the first GPS reading and end with the last GPS reading in the sequence. Consequently, the total trip duration is observed precisely (as the difference of the two time stamps).

The next step is to estimate the route taken in each trip \( i \in \mathcal{I} \), by which we mean the sequence \( R_i = (R_{i,1}, \ldots, R_{i,n_i}) \) of links traversed (so that \( R_{i,k} \) for \( k \in \{1, \ldots, n_i\} \) is an element of the set \( \mathcal{J} \) of network links), the distance \( d_{i,k} \) traversed for each link \( R_{i,k} \) (so that \( d_{i,k} \) is equal to the length of link \( R_{i,k} \) for all except possibly the first and last link of the trip), and the travel time \( T_{i,k} \) on each link \( R_{i,k} \). Obtaining this estimate is called map-matching, a problem for which there are numerous high-quality approaches (Newson & Krumm, 2009; Hunter et al., 2014).

Some of those approaches provide the uncertainty associated with the path and with the link travel times. Although this uncertainty can be taken into account during statistical modeling, we have found little benefit to this approach in prior work on predicting travel time distributions on routes (see Section 5.2 of Westgate et al. (2015)). This is due to the fact that the start and end locations and times for the trips are known with a high degree of accuracy, and the uncertainty is primarily with regards to the allocation of that total time to the links of the route. Ignoring this uncertainty can affect the estimates of the model parameters (Jenelius, 2014).
but in our experience does not have a substantial effect on the predictions for the travel time on the entire trip. For this reason, we use deterministic rather than probabilistic estimates of \( R_{i,k} \), \( d_{i,k} \), and \( T_{i,k} \) as obtained from a standard map-matching procedure (Newson & Krumm [2009]).

Having obtained the values \( T_{i,k} \), we model \( T_{i,k} \) as the ratio of several factors:

\[
T_{i,k} = \frac{d_{i,k}}{E_i S_{i,k}} \quad i \in I, k \in \{1, \ldots, n_i\}
\]

where \( E_i \) and \( S_{i,k} \) are positive-valued latent variables (unobserved quantities) associated with the trip and the trip-link pair, respectively. The latent variable \( E_i \) captures the fact that the trip \( i \) may have, say, 10% faster speeds than average on every link in the trip. This could occur for example due to traffic conditions that affect the entire trip, or to driver habits or vehicle characteristics. The latent variable \( S_{i,k} \) represents the vehicle speed on the link before accounting for the trip effect \( E_i \), and captures variability in speed due to local conditions such as local traffic situations, construction on the link, and short-term variations in driver behavior. The model (1) decomposes the variability of travel time on route \( R_i \) into two types: link-level variability captured by \( S_{i,k} \), and trip-level variability captured by \( E_i \).

We model \( E_i \) with a log-normal distribution:

\[
\log(E_i) \sim N(0, \tau^2)
\]

for unknown variance parameter \( \tau^2 \). The evidence from the Seattle mobile phone data that supports our modeling assumptions is discussed in Section 4. Other data sets may have different characteristics, and the assumption (2) can be replaced if needed with a \( t \) distribution on \( \log(E_i) \) without substantively affecting the computational method described in Section 3 (Liu [1997]).

We model \( S_{i,k} \) in terms of an unobserved discrete congestion state \( Q_{i,k} \in \{1, \ldots, Q\} \) affecting the traversal of link \( R_{i,k} \) in trip \( i \). Notice that this congestion state is allowed to depend on the trip, so that \( Q_{i,k} \) could be different for two trips traversing the same link \( R_{i,k} \) at the same time. This is motivated by features in the data, as we show in Section 4. Assume that the week has been divided into time bins \( b \in B \) that reflect distinct traffic patterns, but do not have to be contiguous (such as “morning rush hour” or “weekend daytime”), and let \( b_{i,k} \) be the time bin during which trip \( i \) begins traversing link \( R_{i,k} \). Conditional on \( Q_{i,k} \), we model \( S_{i,k} \) with a log-normal distribution:

\[
\log(S_{i,k}) | Q_{i,k} \sim N(\mu_{R_{i,k},b_{i,k},Q_{i,k}}, \sigma^2_{R_{i,k},b_{i,k},Q_{i,k}})
\]

where \( \mu_{j,b,q} \) and \( \sigma^2_{j,b,q} \) are unknown parameters associated with travel speed on link \( j \in J \) under conditions \( q \in \{1, \ldots, Q\} \), in time bin \( b \in B \). The normal distribution for \( \log(S_{i,k}) \) can be replaced with a \( t \) distribution, or a skew normal or skew \( t \) distribution, as needed without substantively changing the computational method described in Section 3 (Lee & McLachlan [2013]).

We use a Markov model for the congestion states \( Q_{i,k} \) (motivated by features in the data; Section 4):

\[
\Pr(Q_{i,1} = q) = p_{(0)}^{(0)}(q) \\
\Pr(Q_{i,k} = q | Q_{i,k-1} = \tilde{q}) = p_{R_{i,k},b_{i,k},Q_{i,k}}(q, \tilde{q}) \quad k \in \{2, \ldots, n_i\}; \ q, \tilde{q} \in \{1, \ldots, Q\}
\]

where \( p_{j,b}^{(0)} \) is an unknown probability vector for the initial congestion state for trips starting on link \( j \), and \( p_{j,b} \) is the transition matrix for the congestion state on link \( j \) conditional on the congestion state in the previous link of the trip, during time bin \( b \). This model captures weekly cycles in the tendency of the link to be congested; for example, there may be a high chance of congestion during rush hour.
also provides a second way to capture dependence of travel time across links (in addition to the trip effect \(E_i\)). Our specifications (1)-(4) imply a normal mixture model for \(\log(T_{i,k})\); for instance, when \(k > 1\) and conditioning on \(Q_{i,k-1}\) we obtain

\[
\log(T_{i,k}) \mid Q_{i,k-1} = \tilde{q} \sim \sum_{q=1}^{Q} p_{R_{i,k},b_{i,k}}(\tilde{q}, q) \cdot \mathcal{N}(\log d_{i,k} - \mu_{R_{i,k},b_{i,k},q}, \sigma_{R_{i,k},b_{i,k},q}^2 + \tau^2).
\] (5)

This mixture model captures important features of the data, including heavy right skew of the distributions of the travel times \(T_{i,k}\), and multimodality of the distributions of \(\log(T_{i,k})\); in particular, a mixture of log-normal distributions provides a good approximation to the distribution of vehicle speeds on individual links (see Section 4). In order to enforce the interpretation of the mixture components \(q\) as increasing levels of congestion, we place the restriction \(\mu_{j,b,q-1} \leq \mu_{j,b,q}\) for each \(j \in \mathcal{J}\), \(b \in \mathcal{B}\), and \(q \in \{2, \ldots, Q\}\).

Typically, there are some network links \(j \in \mathcal{J}\) with insufficient data (in terms of the number of link traversals \(m_j \equiv \{|i \in \mathcal{I}, k \in \{1, \ldots, n_i\} : R_{i,k} = j\}||\) to accurately estimate the link-specific parameters \(\mu_{j,b,q}, \sigma_{j,b,q}^2, p_{j,b}^{(0)}\), and \(p_{j,b}\). For such links, we use a single set of parameters within each road category, by which we mean the combination of road classification (e.g., “highway”, “arterial”, or “major road”) and speed limit, and which is denoted by \(c(j)\) for each link \(j\). Defining a minimum number \(m\) of traversals, for links with \(m_j < m\) we set

\[
\mu_{j,b,q} = \mu_{c(j),b,q}, \quad \sigma_{j,b,q}^2 = \sigma_{c(j),b,q}^2, \quad p_{j,b}^{(0)} = p_{c(j),b}^{(0)}, \quad p_{j,b} = p_{c(j),b}
\]

for \(q \in \{1, \ldots, Q\}\), \(b \in \mathcal{B}\), \(j \in \mathcal{J}\) : \(m_j < m\) (6)

where \(\mu_{c,b,q}, \sigma_{c,b,q}^2, p_{c,b}^{(0)}\), and \(p_{c,b}\) are parameters associated with the road category \(c \in \mathcal{C}\).

Our travel time model (1)-(6) incorporates both trip-level variability like driver effects, and link-level variability due for example to construction. It also captures the effect of weekly cycles, speed limit, and road classification. Combined with an assumption regarding changes in vehicle speed across the link, it provides a realistic model for the location of the vehicle at all times during the trip. Since links are typically short, we assume constant speed of each vehicle across the link. This assumption can be relaxed using the approach described in [Hofleitner et al. (2012a)], although those authors find only a modest improvement in predictive accuracy relative to a constant-speed assumption. Our model does not currently take into account real-time traffic conditions, although this is a direction of ongoing work; see Section 5.

### 3 Training and Prediction

Computation is done in two stages: model training (obtaining parameter estimates) and prediction (plugging those estimates into the model to obtain a forecast distribution). Training can be done offline and repeated periodically, incorporating new data and discarding outdated data. This can be used to accumulate information about rarely-observed links and to account for changes in the weekly cycles of travel times, or trends like gradual increases in total traffic volume. Prediction is done at the time when a user makes a routing or travel time query, so must be very computationally efficient.

An alternative Bayesian approach is to take into account uncertainty in the parameter values when doing prediction, by integrating over the posterior distribution of the parameters (their probability distribution conditional on the data). This is typically done using Markov chain Monte Carlo computation (Gelman et al. 2013). However, this approach is more computationally intensive, with complexity that could scale poorly (Woodard & Rosenthal, 2013). Additionally, in other work on travel time distribution prediction we...
have found that the parameter uncertainty is dwarfed by the travel time variability ([Westgate et al.] 2015), so that there is little change in the predictions using the more computationally intensive approach. For notational simplicity, we drop the use of common parameters as in (6).

### 3.1 Training

We train the model using maximum a posteriori (MAP; cf. [Cousineau & Helie (2013)]) estimation, an approach in which one estimates the vector $\theta$ of unknown quantities in the model to be the value that maximizes the posterior density of $\theta$. Latent variables can either be integrated out during this process (if computationally feasible), or included in the vector $\theta$. For example, in image segmentation using Markov random field models, there is a long history of MAP estimation where $\theta$ is taken to include all of the latent variables. In this case the latent variables correspond to the segment associated with each image pixel, of which there can be hundreds of thousands, and approximate MAP estimation of the latent variables provides a computationally tractable solution ([Besag 1986; Grava et al. 2007]). We take an intermediate approach, integrating over the congestion variables $Q_{i,k}$ (for which the uncertainty is high), and doing MAP estimation of the trip effects $E_i$ (for which the uncertainty is lower due to having multiple observations $T_{i,k}$ for each trip). So we take $\theta \equiv \{\tau, \{\mu_{j,b,q}, \sigma_{j,b,q}, P_{j,b}^{(0)}, P_{j,b}\}\}_{j,b \in J, q \in \{1, ..., Q\}, i \in I} (\log E_i)_{i \in I}$ to include the parameters and the trip effects. We are able to do MAP estimation of $\theta$ using an efficient iterative procedure with closed-form updates. Notice that MAP estimation can depend on the transformations chosen in $\theta$ (e.g., the exponentiated MAP estimate of $\log E_i$ is not equal to the MAP estimate of $E_i$); our choice of $\theta$ uses transformations that are mathematically convenient and yield good results. Additionally, our computational procedure is guaranteed to obtain only a local maximum of the posterior density, but by repeating the procedure multiple times using random initializations one can increase the chance of finding the global maximum.

MAP estimation requires specification of a prior density for the parameters; we use the prior

$$p(\theta; \{\mu_{j,b,q}, \sigma_{j,b,q}, P_{j,b}^{(0)}, P_{j,b}\}) \propto 1$$

that is uniform on the support of the parameter space. This prior is non-integrable, but leads to valid MAP estimation. Such uniform priors on unbounded parameter spaces are commonly used in situations where there is little or no prior information regarding the parameter values (Section 2.8 of [Gelman et al. 2013]); see for instance [Gelman (2006)] for the use of uniform priors for standard deviation parameters like $\sigma_{j,b,q}$.

Now consider the observed data to consist of the transformed values $\{\log \tilde{S}_{i,k}\}_{i \in I, k \in \{1, ..., n_i\}}$ where $\log \tilde{S}_{i,k} \equiv \log d_{i,k} - \log T_{i,k}$ is the log speed during link traversal $i, k$. Because the prior density is uniform, MAP estimation of $\theta$ corresponds to maximizing the product of the likelihood function, multiplied by the probability density of the trip effects:

$$p(\theta; \{\log \tilde{S}_{i,k}\}) = p(\{\log \tilde{S}_{i,k}\}) p(\theta) / p(\{\log \tilde{S}_{i,k}\})$$

$$\propto p(\{\log \tilde{S}_{i,k}\}) p(\{\log E_i\} | \tau)$$

(7)

over $\theta$. Maximum likelihood estimation of $\theta$ would maximize $p(\{\log \tilde{S}_{i,k}\} | \theta)$; by including the second term $p(\{\log E_i\} | \tau)$ in our objective function (7), we reduce noise in the estimated log $E_i$ values (a technique called regularization; [James et al. 2013]). Notice that to compute the likelihood function $p(\{\log \tilde{S}_{i,k}\} | \theta) = \sum_{Q_{i,k}} p(\{Q_{i,k}, \log \tilde{S}_{i,k}\} | \theta)$, one must sum over all the possible values of the latent variables $\{Q_{i,k}\}$.

Expectation Maximization (EM) is an iterative method for maximum likelihood or MAP estimation in the presence of many latent variables; accessible introductions are given in [Hofleitner et al. 2012a] and, in more detail, [Bilmes 1997]. EM is most efficient when the parameter updates in each iteration can be done in closed form, which is not the case in our model. However, we can obtain closed-form updates using a variant called Expectation Conditional Maximization (ECM; [Meng & Rubin 1993]). ECM allows for
closed-form updates in situations where the parameter vector can be partitioned into subvectors, each of which would have a closed-form EM update if the remaining parameters were known.

We apply ECM by partitioning the parameter vector into the two subvectors
\[ \theta_1 = (\tau, \{ \phi_{j,b,q}, \sigma_{j,b,q}, p_{j,b}^{(0)} \}) \quad \text{and} \quad \theta_2 = \{ \log E_i \}. \]
In ECM, attention focuses on the complete-data log posterior density \( \log p(\theta | \{ Q_{i,k}, \log \tilde{S}_{i,k} \}) \), which is equal to a term that does not depend on \( \theta \), plus
\[
\log p(\{ Q_{i,k}, \log \tilde{S}_{i,k} \} | \theta) + \log p(\{ \log E_i \} | \tau) = \\
\sum_{i \in I} \log(p(0)_{R_i,1,b_1}(Q_{i,1})) + \sum_{i \in I, k \in \{2, \ldots, n_i\}} \log(p_{R_i,k,b_k}(Q_{i,k-1}, Q_{i,k})) \\
+ \sum_{i \in I, k \in \{1, \ldots, n_i\}} \left[ \frac{-\log \sigma_{R_i,k,b_k,q_{i,k}}^2}{2} - \frac{(\log \tilde{S}_{i,k} - \log E_i - \mu_{R_i,k,b_k,q_{i,k}})^2}{2\sigma_{R_i,k,b_k,q_{i,k}}^2} \right] \\
+ \sum_{i \in I} \left[ \frac{-\log \tau^2}{2} - \frac{(\log E_i)^2}{2\tau^2} \right]
\]
(8)

To update a parameter subvector \( \theta_j \) in each iteration of ECM, one treats the remaining subvectors \( \theta_{[-j]} \) as fixed, and maximizes over \( \theta_j \) the expectation of (8) with respect to \( p(\{ Q_{i,k} \} | \theta_{\text{cur}}, \{ \log \tilde{S}_{i,k} \}) \). The latter quantity is the probability distribution of the congestion variables conditional on the data and the current value \( \theta_{\text{cur}} \) of \( \theta \). Such an update leads to an increase in the value of the objective function (7) in each iteration, and ultimately to a (local) maxizer of (7), as shown in Meng & Rubin (1993).

This yields the following procedure in each iteration. The first step is to run the forward-backward algorithm for each trip. Briefly, the model (4) is a one-dimensional Markov model for \( \{ Q_{i,k} : k \in 1, \ldots, n_i \} \) for each trip \( i \). Also, the model (1)-(3) implies a conditional distribution for the observed data \( \log(\tilde{S}_{i,k}) \) given \( Q_{i,k} \), namely \( \log(\tilde{S}_{i,k}) | Q_{i,k}, \log E_{i,\text{cur}} \sim N(\log E_{i,\text{cur}} + \mu_{R_{i,k},b_{i,k},q_{i,k}}, \sigma_{R_{i,k},b_{i,k},q_{i,k}}^2). \) So \( p(\{ Q_{i,k} \} | \theta_{\text{cur}}, \{ \log \tilde{S}_{i,k} \}) \) is a hidden Markov model for each \( i \), for which computation can be done exactly and efficiently using the forward-backward algorithm (cf., Russell & Norvig 2009). Use the forward-backward algorithm to calculate the values of \( \phi_{i,k}(q) \equiv \Pr(Q_{i,k} = q | \theta_{\text{cur}}, \{ \log \tilde{S}_{i,k} \}) \) and \( \psi_{i,k}(\tilde{q}, q) \equiv \Pr(Q_{i,k-1} = \tilde{q}, Q_{i,k} = q | \theta_{\text{cur}}, \{ \log \tilde{S}_{i,k} \}) \) over \( i \in I, k \in \{1, \ldots, n_i\} \) and \( \tilde{q}, q \in \{1, \ldots, \mathcal{Q}\}. \)

The second step is to update the link parameter and \( \tau \) estimates as:
\[
\mu_{j,b,q}^{\text{new}} = \frac{\sum_{i \in I, R_i = j, b_i = b} \phi_{i,k}(q)(\log \tilde{S}_{i,k} - \log E_{i,\text{cur}})}{\sum_{i \in I, R_i = j, b_i = b} \phi_{i,k}(q)} \\
\sigma_{j,b,q}^{\text{new}} = \frac{\sum_{i \in I, R_i = j, b_i = b} \phi_{i,k}(q)(\log \tilde{S}_{i,k} - \log E_{i,\text{cur}} - \mu_{j,b,q}^{\text{new}})^2}{\sum_{i \in I, R_i = j, b_i = b} \phi_{i,k}(q)} \\
p^{(0)}_{j,b}^{\text{new}}(q) = \left( \sum_{i : R_{i,1} = j, b_{i,1} = b} \phi_{i,1}(q) \right) / \left( \sum_{i : R_{i,1} = j, b_{i,1} = b} 1 \right) \\
p_{j,b}^{\text{new}}(\tilde{q}, q) = \left( \sum_{i : R_{i} = j, b_{i} = b, k > 1} \psi_{i,k}(\tilde{q}, q) \right) / \left( \sum_{i : R_{i} = j, b_{i} = b, k > 1} \phi_{i,k-1}(\tilde{q}) \right) \\
\tau^{\text{new}} = \frac{1}{|I|} \sum_{i \in I} (\log E_{i,\text{cur}})^2.
\]
(9)

Third, recalculate the values of \( \phi_{i,k}(q) \) and \( \psi_{i,k}(\tilde{q}, q) \) and update the trip effect estimate as follows, where
\[ a_{i,k} = \sum_{q=1}^{Q} \frac{\phi_{i,k}(q)}{\sigma_{R_{i,k}, b_{i,k}, q}} \] and \[ h_{i,k} = \sum_{q=1}^{Q} \frac{\phi_{i,k}(q) \mu_{R_{i,k}, b_{i,k}, q}}{\sigma_{R_{i,k}, b_{i,k}, q}}. \]

\[ \log E_{i}^{\text{new}} = \frac{\sum_{k \in \{1, \ldots, n_{i}\}} (a_{i,k} \log \hat{S}_{i,k} - h_{i,k})}{1/\tau_{\text{curr}} + \sum_{k \in \{1, \ldots, n_{i}\}} a_{i,k}}. \] 

We iterate until there is no change in the parameter estimates up to the third significant figure. The time complexity of each iteration of the ECM algorithm, implemented without parallelization, is \( O\left(Q^2(|J| \times |\mathcal{B}| + \sum_{i \in \mathcal{J}} n_{i})\right) \), i.e. the procedure is linear both in the number of network links \(|J|\) and in the average number of traversals per link \((\sum_{i \in \mathcal{J}} n_{i})/|J|\). The latter is a measure of the spatial concentration of data. The number of ECM iterations required for convergence is also likely to grow with the size of the network and the spatial concentration of data, but this change in efficiency is difficult to characterize in general. In practice, ECM often converges quickly; for the Seattle case study of Section 4, fewer than 50 iterations are required.

For the case study, training takes about 15 minutes on a single processor. The road map of North America has about 400 times as many directional links as the Seattle metropolitan region, so it would take roughly 4.2 days to run our implementation for the continent of North America, using the same number of iterations of ECM that we used in Seattle. This time can be reduced dramatically, since each of the parameter updates (9)-(10) can be computed using parallelization across trips and/or links. Since training can be done offline, and the model only needs to be retrained occasionally (e.g., twice a month), the training procedure is expected to be sufficiently fast for commercial mapping services (which operate at a continental scale). Even if the spatial concentration of data is increased by several orders of magnitude (e.g., where smartphone-based GPS systems provide coverage for a significant portion of drivers) the training procedure is likely to be efficient enough for commercial use by exploiting parallelism.

The updates of \( p_{j,b,q}, \sigma_{j,b,q} \), and \( p_{j,b}^{(0)} \) in (9) are the same as those in EM for normal mixture models (Bilmes 1997), except that we restrict the calculation to relevant subsets of the data and adjust for the estimated trip effect \( \log E_{i}^{\text{curr}} \). The update of \( p_{j,b} \) is analogous to that of \( p_{j,b}^{(0)} \), but for a transition matrix instead of a probability vector. The update of \( \tau \) is the maximum likelihood estimate conditional on \( \{\log E_{i}^{\text{curr}}\}_{i \in \mathcal{J}} \). The update of \( \log E_{i} \) in (10) is not a standard form. However, in the special case where the \( \sigma_{j,b,q}^{2} \) are equal for all \( j, b, q \), and \( q \), the updated value \( \log E_{i}^{\text{new}} \) is approximately the average across \( k \) of the difference between \( \log \hat{S}_{i,k} \) and its expectation under the model, which is a reasonable estimator for the trip effect.

### 3.2 Prediction

Prediction in model (1)-(4) is done by Monte Carlo simulation. For a specified route \( R_{i} \), starting at a specified time, we simulate vectors of travel times \( (T_{i,1}, \ldots, T_{i,n_{i}}) \) directly from the model, using the trained parameter values. During prediction the time bins \( b_{i,k} \) for \( k > 1 \) are not fixed, but rather depend on the simulated value of \( (T_{i,1}, \ldots, T_{i,k-1}) \). Having obtained simulated values from the distribution of total travel time \( \sum_{k=1}^{n_{i}} T_{i,k} \) in this way, Monte Carlo is used to approximate the expectation, quantiles, percentiles, and other summaries of that distribution.

The time complexity of prediction, without parallelization, is \( O(M \times n_{i}) \) where \( M \) is the number of Monte Carlo samples. Importantly, it does not depend on the size of the road network \(|J|\), and can be further reduced by doing the simulations in parallel. For the analysis of Section 4 we use 1000 Monte Carlo simulations per route; additional simulations yielded no noticeable improvements in predictive accuracy. Prediction took an average of only 17 milliseconds per route, on a single processor. Commercial mapping services require computation times of less than 50 milliseconds for a route recommendation, and several
orders of magnitude less than that for a single route query that is used as part of a sophisticated optimization like ranking points of interest by driving time. So the running time of our method even on a single processor is fast enough for some purposes, and when combined with parallelism would be efficient enough for more complex queries involving multiple routes.

4 Seattle Case Study

We obtained anonymized mobile phone GPS data gathered from Windows phones, for the Seattle metropolitan region during a time window in 2014 (Figure 1). No personal identifiers were available. We isolated vehicle trips and estimated the corresponding routes as described in Section 2, yielding 150,766 trips that have distance along the road network of at least 3 km. These trips have mean trip duration of 800 s, maximum trip duration of 8555 s, mean trip distance of 11 km, and maximum trip distance of 105.8 km.

In Figure 2 we show the volume of trips by the hour of the week in which the trip began. The volume dips overnight, and peaks associated with morning and evening rush hour are present on weekdays. The volume of recorded trips is highest on Saturday and Sunday daytimes, most likely due to higher usage of GPS-utilizing phone applications on weekends. We define the time bins \( b \in B \) based in part on the changes in volume over the week as seen in Figure 2 yielding five bins: “AM Rush Hour”: weekdays 7-9 AM; “PM Rush Hour”: weekdays 3-6 PM; “Nighttime”: Sunday-Thursday nights 7 PM-6 AM, Friday night 8 PM-9 AM, and Saturday night 9 PM-9 AM; “Weekday Daytime”: remaining times during weekdays; and “Weekend Daytime”: remaining times during weekends.

There are 221,980 directional network links in the study region. Of the link traversals in our dataset, 32.8% are on highways, 34.8% are on arterials, 14.0% are on major roads, 12.1% are on streets, and 6.3% are on other road classifications. We take the minimum number of observations per parameterized link (as used in (6)) to be \( m = 200 \), since numerous authors have used sample sizes in the hundreds to successfully fit low-dimensional normal mixture models (Banfield & Raftery, 1993; Bensmail et al., 1997). There are 10,435 directional links in the Seattle area road network that satisfy this criterion. Although this is only 4.7% of the links in the network, they account for 67.3% of link traversals.

Using two congestion states \( (Q = 2) \), in Figure 3 we validate the estimated distribution of travel time during evening rush hour for some road links, based on our trained model. We focus on the links that have the highest number of observed travel times \( m_j \), showing the two most commonly observed highway links (omitting links on the same section of highway), the most commonly observed “arterial” link, and the most
commonly observed “major road” link. For each of these links, Figure 3 gives the histogram of the log travel times during evening rush hour from the training data, adjusted for the estimated trip effect (i.e., the histogram of \( \log T_{i,k} - \log E_i \)). We also overlay curves showing the corresponding estimated density from our model (i.e., \( \sum_{q=1}^{Q} P_{R_{i,k,b_{i,k}},q}(0)(\log d_{i,k} - \mu_{R_{i,k,b_{i,k},q}}, \sigma_{R_{i,k,b_{i,k},q}}^2) \)). The travel time distributions are so heavily right-skewed that patterns are easier to see when plotted on the log scale as we do here.

Figure 3 illustrates the multimodality of the distribution of log travel time. Histograms restricting to particular 15-minute time periods have a similar shape to those in Figure 3, indicating that the multimodality is not caused by aggregating over time periods. The mixture of log-normals used by TRIP appears to fit the observations well. By contrast, assuming a single gamma, normal, or log-normal distribution for travel times in a particular time period, as done for example in Hofleitner et al. (2012a,b); Westgate et al. (2013) and Hunter et al. (2013a), leads to poor model fit for this mobile phone data.

Next we motivate use of the Markov model (4) for the congestion states \( Q_{i,k} \), and our choice to allow \( Q_{i,k} \) to depend on the trip rather than just the link and time. First, we will give evidence that the auto-correlation of log travel times within a trip is high and decreases with distance. Second, we show that the correlation of log travel times for vehicles traversing the same link at roughly the same time is not consistently high. These observations suggest that it is more appropriate to model the congestion level as a property of the individual trip, rather than as a property of the link and time. They also suggest the use of a Markov model for \( Q_{i,k} \), which can capture association that decays with distance.

Our example corresponds to the first 10 links of the second route shown in Figure 4 (highway 520 West). In Figure 5, we illustrate the correlation of log travel times within and across trips on this sequence of links. The plots in the left column show that the autocorrelation of log travel times within the same trip is high and decreases with distance. The plots in the middle column show that the correlation of log travel times is not consistently high for pairs of distinct trips traversing the sequence of links in the same 15-minute time period. Although the correlation appears high in one of the two plots in the middle column, the scatterplots in the right column show that the travel times do not have a strong association across trips. In summary, the congestion level experienced appears to depend on the trip, and not just the links traversed.

This effect occurs in part due to a HOV lane on this section of highway, so that vehicles traveling in the HOV lane experience less congestion than other vehicles. Additionally, this section of 520 West is just before the interchange with highway 405. The traffic can be congested on 405, which can cause lane-
Figure 4: Three routes in the road network and their predicted travel time distributions during evening rush hour, compared to observed travel times from the test data. Left: the routes; right: histograms of the observed travel times and curves showing the predictive densities. Top: a 3.4 km section of highway 405 North; middle: a 3.3 km route consisting of a section of 520 West followed by the exit ramp to 405 South; bottom: a 0.9 km section of northbound Lake City Way NE.

Having motivated our modeling choices, we now evaluate predictive accuracy. The data from the first 3/4 of the study time period were used to train the various models, and those from the last 1/4 of the study time period were used to test predictive accuracy. Figure 4 shows three of the most common routes in the road network: two highway routes and one arterial route. Histograms of the observed travel times for these routes in the test data during evening rush hour are shown, along with the predicted probability density of evening rush hour travel time obtained from TRIP. The observed travel times are obtained from all trips that traverse the entire route of interest; these typically are trips that have longer routes, but for which we focus on the travel time only for the route of interest. The predictive densities match the histograms well, capturing the heavy right skew and multimodal nature of the travel times, and accurately predicting the amount of variability of the distributions.

Next we evaluate predictive accuracy on the entire test dataset (36,493 trips on routes throughout the network), comparing TRIP to several alternatives. First, we compare to three simplified versions of TRIP that drop one or both of the types of dependence across links. One type of dependence is induced by the trip effect $E_i$, so we consider dropping this term from the model. The other type is caused by the Markov model on $Q_{i,k}$, so we consider replacing this with an independence model, $\Pr(Q_{i,k} = q) = p^{(0)}_{R_{i,k}, b_{i,k}}(q)$ for all $k \in \{1, \ldots, Q\}$. 

specific congestion on 520 West extending significant distances from congested exits. Such effects from HOV lanes and congested interchanges are common throughout the road network, so that vehicles traveling on the same link in the same time period (per GPS resolution) can experience very different congestion levels depending on the choice of lane. Moreover, the choice of lane can be a function of the intended route.
Second, we compare the results to inferences from an adaptation of Clearflow ([Microsoft Research, 2012]). Clearflow models the distribution of travel time on each link in the network using Bayesian structure search and inference to integrate evidence from a large number of variables, including traffic flow sensor measurements, speed limit, road classification, road topological distinctions, time of day, day of week, and proximity to schools, shopping malls, city centers, and parks. It was designed for accurate prediction of mean real-time flows on all links of large metropolitan traffic networks. In practice, the Clearflow inferences about flows on individual links are used to guide route planning procedures that generate routes via summing the times of the set of links of a trip between starting point and destination. For this reason Clearflow was not targeted at modeling the distribution of travel time on entire routes. However, we can combine Clearflow with an assumption of independence across links, in order to produce distribution predictions on routes for the purposes of comparison.

Finally, we compare to a regression approach that models the travel time for the entire trip, like methods for predicting ambulance travel times ([Budge et al., 2010] [Westgate et al., 2015]). In particular, we use a linear regression model for log trip travel time in terms of: (a) the log of route distance; (b) the time bin \( b_{i,1} \) in which the trip begins; and (c) the log of the travel time according to the speed limit. We include an interaction term between (b) and (c), meaning that the linear slope for (c) is allowed to depend on the value of (b). The assumptions of the linear regression model hold approximately; for example, scatterplots of the log travel time and the variables (a) and (c) show an approximately linear relationship.

A limitation of both Clearflow and linear regression is that they cannot be used to update the travel time prediction en route, taking into account the same vehicle's speed on the completed part of the route (e.g., if the vehicle drove the first half of the route more quickly than expected given the current measured conditions, one should predict the speed on the second half of the route to be higher than expected). With TRIP, the vehicle travel times \( T_{i,k} \) on the completed part of the route provide information regarding the trip effect \( E_i \), and regarding the congestion variables \( Q_{i,k} \) on the remaining part of the route. This yields an
informed predictive distribution for the travel time on the remainder of the route.

Table 1 and Figure 6 report the accuracy of deterministic and interval predictions of travel time on the test data. To obtain a deterministic prediction, we use the geometric mean of the travel time distribution. This is more appropriate than the arithmetic mean, due to the heavy right skew of travel time distributions. To obtain an interval prediction with theoretical coverage \(100(1-\alpha)\) where \(\alpha \in (0, 1)\), we take the lower and upper bounds of the interval to be the \(100(\alpha/2)\) and \(100(1-\alpha/2)\) percentiles of the travel time distribution (for example, the 95% interval is the 2.5 and 97.5 percentiles).

In Table 1 we report the geometric mean across trips of the percentage error of deterministic predictions (the absolute difference between the predicted and observed travel time, divided by the observed travel time and multiplied by 100). We also report the mean absolute error. Third, we report the bias of the deterministic prediction on the log scale: precisely, the mean across trips of the log of the prediction minus the log of the observed travel time. The bias does not measure how far the predictions are from the observations, so a small amount of nonzero bias is not a problem. However, adjusting the deterministic predictions to remove the bias can sometimes improve the accuracy as measured using the percentage error or the mean absolute error. Specifically, we also report those accuracy measures after multiplying the deterministic prediction by the exponent of -1 times the log-scale bias. Such a bias adjustment is also done by Westgate et al. (2015).

To measure the accuracy of interval predictions, we first report their empirical coverage, meaning the percentage of test trips for which the observed travel time is inside of the predictive interval. If the variability is predicted correctly, the empirical coverage is close to the theoretical coverage. Additionally, we report the average width of the interval. If two methods both have, say, 95% interval predictions with 95% empirical coverage, the method with the narrower intervals is preferred. This is because both methods are accurately characterizing variability, but one is taking better advantage of the information in the data to give a narrow predicted range (Gneiting et al., 2007).

As seen in Table 1, the bias is small for all of the methods (the largest being .037, which corresponds to a factor of \(\exp\{.037\} = 1.038\) in travel time, i.e. a bias of 3.8%). The percentage error of deterministic predictions is similar for all the methods except linear regression, with the predictions differing from the actual values by 10.1-11.3% for all the methods except linear regression, which has error of over 13%. The
Table 1: Accuracy of deterministic predictions for the Seattle test data.

<table>
<thead>
<tr>
<th>Accuracy Measure</th>
<th>TRIP</th>
<th>TRIP, no trip effect</th>
<th>TRIP, no Markov model</th>
<th>TRIP, no dependence</th>
<th>Clearflow</th>
<th>Linear regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometric mean of [predicted - actual]/actual</td>
<td>11.2%</td>
<td>10.6%</td>
<td>11.3%</td>
<td>10.9%</td>
<td>11.0%</td>
<td>13.2%</td>
</tr>
<tr>
<td>Geometric mean of [predicted - actual]/actual after bias correction</td>
<td>10.4 %</td>
<td>10.2%</td>
<td>10.4%</td>
<td>10.5%</td>
<td>10.1%</td>
<td>13.3%</td>
</tr>
<tr>
<td>Mean absolute error (s)</td>
<td>136.5</td>
<td>135.0</td>
<td>136.2</td>
<td>135.3</td>
<td>133.2</td>
<td>154.1</td>
</tr>
<tr>
<td>Mean absolute error after bias correction (s)</td>
<td>132.7</td>
<td>133.0</td>
<td>132.6</td>
<td>134.5</td>
<td>129.7</td>
<td>154.4</td>
</tr>
<tr>
<td>Bias on log scale</td>
<td>.034</td>
<td>.023</td>
<td>.035</td>
<td>.029</td>
<td>.037</td>
<td>-.007</td>
</tr>
</tbody>
</table>

same conclusions hold after bias correction, and when considering mean absolute error instead of percentage error. The accuracy of TRIP is nearly as good as Clearflow, despite the fact that Clearflow harnesses a larger number of variables than those considered by TRIP in its current form.

Figure 6 shows that, out of the methods considered, only TRIP and linear regression have predictive interval coverage that is close to correct. The other methods have dramatically lower coverage, Clearflow’s coverage of 95% intervals being 69.6%. Clearflow and the simplest version of TRIP have similar coverage because they both assume independence of travel time across the links of a trip. For the simplified versions of TRIP that incorporate one out of the two kinds of dependence, the coverage is much better than that of the methods that assume independence, but still well below the desired value. Figure 6 also shows that the interval predictions from TRIP are 13-16% narrower on average than those from linear regression, evidence that TRIP is substantially better for interval prediction.

5 Conclusions

We have introduced a method (TRIP) for predicting the probability distribution of travel time on arbitrary routes in the road network, at arbitrary times. We evaluated TRIP on a case study utilizing mobile phone data from the Seattle metropolitan region. Based on a complexity analysis and on the running time of the algorithms for this case study, we argue that TRIP is likely to be computationally feasible for the continental-scale road networks and high-volume data of commercial mapping services.

While TRIP did not provide as accurate deterministic predictions of travel time as Microsoft’s commercial system (Clearflow), we found the interval predictions from TRIP to be far better. Clearflow’s consideration of flows on a segment by segment basis is valuable for use with current route planning procedures, which consider the road speeds on separate segments in building routes. However, such independent handling of inferences about segments can lead to underprediction of route-specific variability. TRIP solves this issue by accurately capturing dependencies in travel time across the links of the trip. Although a linear regression approach yields reasonable accuracy of interval predictions, it gives poor deterministic predictions. To our knowledge TRIP is the first method to provide accurate predictions of travel time reliability for complete, large-scale road networks.

We plan to extend TRIP to incorporate additional variables including those used in Clearflow learning and inference. For example, this would allow TRIP to take into account real-time information about traffic conditions, as measured using data from sensors installed in highways, or average measured GPS speeds from mobile phones during the current time period. This extension has the potential to provide much
narrower distribution forecasts and predictive intervals, and even more accurate deterministic estimates.

We also see valuable work ahead in developing or adapting efficient route-planning procedures to consider variability of whole routes in the construction of trips. TRIP predicts travel time variability on entire routes; it is not straightforward to employ these inferences in segment-focused route planning algorithms like Dijkstra or A* search. There are opportunities for using a hybrid of procedures focused on mean travel time on segments with filtering and prioritization based on variability. Variability of alternate routes generated by planners in a segment-focused manner might be displayed to consumers so decisions could be made in accordance with risk preferences.

Finally, there is opportunity to employ active information gathering methods to guide both selective real-time sensing of different portions of a road network and the bulk collection of data to reduce uncertainty about the flows over segments and routes. There has been related prior work on the use of active sensing for reducing uncertainty about the travel time on segments in a demand-weighted manner [Krause et al., 2008]. The work considers the probabilistic dependencies across a city-wide traffic network and the value of sensing from different regions for reducing uncertainty across the entire road network. We foresee the use of similar methods in combination with TRIP to guide the optimal collection of data.

References


James, G., Witten, D., Hastie, T., and Tibshirani, R. *An Introduction to Statistical Learning with Applications in R*. Springer, New York, 2013.


