Arriving on time: estimating travel time distributions on large-scale road networks

Timothy Hunter Aude Hofleitner Jack Reilly Walid Krichene Jérôme Thai
UC Berkeley
{tjhunter, aude.hofleitner, jackdreilly, walid, jerome.thai}@berkeley.edu

Anastasios Kouvelas
kouvelas@berkeley.edu

Pieter Abbeel
pabbeel@berkeley.edu

Alexandre Bayen
bayen@berkeley.edu

ABSTRACT

Most optimal routing problems focus on minimizing travel time or distance traveled. Oftentimes, a more useful objective is to maximize the probability of on-time arrival, which requires statistical distributions of travel times, rather than just mean values. We propose a method to estimate travel time distributions on large-scale road networks, using probe vehicle data collected from GPS. We present a framework that works with large input of data, and scales linearly with the size of the network. Leveraging the planar topology of the graph, the method computes efficiently the time correlations between neighboring streets. First, raw probe vehicle traces are compressed into pairs of travel times and number of stops for each traversed road segment using a 'stop-and-go' algorithm developed for this work. The compressed data is then used as input for training a path travel time model, which couples a Markov model along with a Gaussian Markov random field. Finally, scalable inference algorithms are developed for obtaining path travel time distributions from the composite MM-GMRF model. We illustrate the accuracy and scalability of our model on a 505,000 road link network spanning the San Francisco Bay Area.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

estimation, machine learning, inference

1. INTRODUCTION

A common problem in trip planning is to make a given deadline, for example arriving at the airport within 45 minutes. Most routing services available today minimize the expected travel time, but do not provide the most *reliable* route, which accounts for the variability in travel times. Given a time budget, a routing service should provide the route with highest probability of on-time arrival, as posed in stochastic on-time arrival (SOTA) routing [14]. Such an algorithm requires the estimation of the *statistical distributions* of travel times, or at least of their means and variances, as done in [11]. Today, only few traffic information platforms are available for the arterial network (the state of the art for highway networks is more advanced) and they do not provide the *statistical distributions* of travel times. The main contribution of the article is precisely to addresses this gap: we present a scalable algorithm for learning path travel time distributions on the entire road network using probe vehicle data.

The increasing penetration rate of probe vehicles provides a promising source of data to learn and estimate travel time distributions in arterial networks. At present, there are two general trends in estimation of travel times using this probe data. One trend, from kinematic wave theory (see [18, 7]), derives analytical probability distributions of travel times and infer their parameters with probe vehicle data. These approaches are computationally intensive, which limits their applicability for large scale networks. The other trend, seen in large-scale navigation systems such as Google Maps, provides coarser information, such as expected travel time, but can scale to world-sized traffic networks.

We bridge the two trends by creating a travel time estimator that (i) provides full probability distributions for arbitrary paths in real-time (sub-second), (ii) works on networks the size of large cities (and perhaps larger) (iii) and accepts an arbitrary amount of input probe data. The model uses a data-driven model which is able to leverage physical insight from traffic flow research. Data-driven models, using dynamic Bayesian networks [6], nearest neighbors [16] or Gaussian models [17] show the potential of such methods to make accurate predictions when large amounts of data are available.

The main physical insights modeled in the article are described in the following paragraphs. First, arterial traffic is characterized by important travel time variability amongst users of the network (Figure 1). This variability is mainly due to the presence of traffic lights and other impediments such as pedestrian crossings and garbage trucks which cause a fraction of the vehicles to stop while others do not. Arterial traffic research [7] suggests that the detection of stops explains most of the variability in the travel time distribution and underline the multi-modality of the distributions.

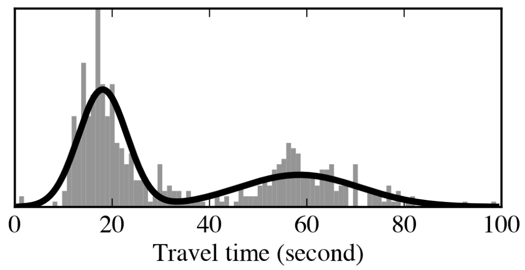

Figure 1: Histogram of travel times collected on a link fit (solid line) using a mixture of Gaussian distributions.

Second, the number of stops on a trajectory exhibits strong spatial and temporal correlations. Traffic lights on major streets may be synchronized to create "green waves": a vehicle which does not stop on a link is likely to not stop on the subsequent link. A different vehicle arriving 10 seconds later may hit the red light on the first link and have a high probability to stop on the subsequent link. This phenomenon is analyzed in [13] using a Markov model with two modes ("slow" and "fast").

Third, besides the the number of stops, travel times may be correlated for the following reasons: (i) the behavior of individual drivers may be different: some car may travel notably faster than some others, (ii) congestion propagates on the network, making neighboring links likely to have similar congestion levels. If a driver experiences a longer than usual travel time on a link because of heavy traffic, he / she can will likely experience heavy traffic on the subsequent links.

We leverage these insight to develop the traffic estimation algorithm presented in this article: an end-to-end, scalable model for inferring path travel time distributions, referred to as a "pipeline" (see Figure 2). It consists of a *learning algorithm* and an *inference algorithm*.

The *learning algorithm* consists in the following steps.

- Section 2: a *Stop-&-Go* filter algorithm detects the number of stops on a link and compresses the GPS traces[1] into values of travel times on traversed links and corresponding number of stops.
- Section 3.1: a *Markov model* (MM) captures the correlations of stopping / not stopping for consecutive links. It characterizes the probability to stop / not stop on a link given that the vehicle stopped or did not stop on the previous link traversed. The *Stop-&-Go* filter produces a set of labeled data to train the Markov model.
- Section 3.2: a *Gaussian Markov Random Field* (GMRF) captures the correlations in travel times between neighboring links, given the number of stops on the links. There is a significant body of prior work in the field of learning with graphical models [10, 9], especially for learning problems on sparse GMRFs [5, 12]. Most of these algorithms do not scale linearly with respect to the dimension of the data, and are unsuitable for very large problems (hundreds of thousands of variables). In particular, it becomes practically impossible to store the entire covariance matrix, so even classical sub-gradient methods such as [3] would require careful engineering.

We exploit the (near) planar structure of the underlying graphical model to more efficiently obtain (approximate) algorithms that scale *linearly* with the size of the network. Our algorithm leverages efficient algorithms to compute the Cholesky decomposition of the adjacency matrix of planar graphs [2]. Our results can be extended to other physical systems with local correlations.

[1]Before feeding raw GPS points to the *Stop-&-Go* filter, each coordinate is mapped to a link and an offset distance from the beginning of the link. We use an efficient path-matching and path-inference algorithm developed in [8].

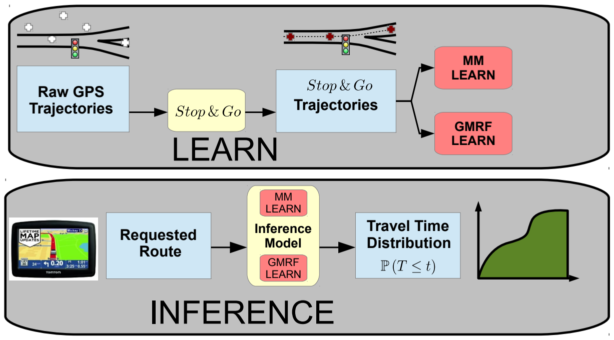

Figure 2: Pipeline of the travel-time estimation model. The learning runs offline and finds optimal parameters for the Markov model and GMRF models. The inference runs online and uses the learned parameters to produce travel time distributions on input paths.

After the learning, we proceed to the *inference algorithm*: we compute the travel time distributions for arbitrary paths in the network (Section 4). While exact inference on this model is intractable due to the number of possible states, we exploit the underlying structure of the graphical model and use a specialized sampling method to obtain an efficient inference algorithm.

Section 5 illustrates the accuracy and scalability of our model on a 505,000 road link network spanning the San Francisco Bay Area.

Our code, as well as a showcase of the model running on San Francisco, is available at `http://traffic.berkeley.edu/navigateSF`.

2. *STOP-&-GO MODEL FOR VEHICLES TRAJECTORIES*

The stops due to traffic signals and other factors (double parking, garbage trucks and so on) represent one of the main source of variability in urban travel times. More generally, consider that a link can have m different discrete states. For a vehicle traveling on link l, the *state* $s_l \in \{0, m-1\}$ of the trajectory is defined as the number of stops on the link. The following algorithm estimates the number of stops given a set of noisy GPS samples from the trajectory on link l. We consider a generic trajectory on a generic link and drop indices referring to the trajectory and link for notation simplicity.

The trajectory of the vehicle is represented by an offset function $T : [0, \tau] \to \mathbb{R}_+$, representing the distance from the beginning of the link to the location of the vehicle at time t. The noisy GPS observations are defined by the times $0 = t_0 < \cdots < t_J \leq \tau$, and the corresponding offsets $x_j = T(t_j) + \epsilon_j$, where $\epsilon_j \sim \mathcal{N}(0, \sigma^2)$ are independent and identically distributed zero mean Gaussian random variables representing the GPS noise.

We process the observations $(t_j, x_j)_{j=0,\dots,J}$ to obtain *stop and go* trajectories of the probe vehicles: the trajectory of a vehicle alternates between phases of *Stop* during which the velocity of the vehicle is zero and *Go* during which the vehicle travels at positive speed. The number of *Stop* phases represents the state of the trajectory. We assume that the sampling frequency is high enough that the speed between successive observations (t_j, x_j) and (t_{j+1}, x_{j+1}) is constant[2] and denoted v_j. Note that speeds are rarely provided by GPS devices or are too noisy to be valuable for estimation.

Maximizing the log-likelihood of the observations is equivalent to solving the following optimization problem

$$\underset{(v_j)_{j\in\{0,\dots,J-1\}}}{\textbf{minimize}} \frac{1}{2}\sum_{j=0}^{J-1}\left(x_{j+1} - x_0 - \sum_{j'=0}^{j} v_{j'}(t_{j'+1} - t_{j'})\right)^2$$

This is a typical least-square optimization problem, which we conveniently write in matrix form as:

$$\underset{v}{\textbf{minimize}}\frac{1}{2}||Av - b||_2^2, \tag{1}$$

with the notation $v = (v_j)_{j\in\{0,\dots,J-1\}}$, $b = (x_j - x_0)_{j\in\{1,\dots,J\}}$ and A is the lower triangular matrix whose entry on line $i \in \{0,\dots,J-1\}$ and column $k \in \{0,\dots,i-1\}$ is given by $t_{k+1} - t_k$.

To detect the *Stop* phases, we add a l_1 regularization term, with parameter λ to Problem (1). The resulting optimization problem is known as the LASSO [15] and reads

$$\underset{v}{\textbf{minimize}}\frac{1}{2}||Av - b||_2^2 + \lambda||v||_1, \tag{2}$$

The solution of (2) is typically sparse [15], which means that there is a limited number of non-zero entries, corresponding to the *Go* phases of the trajectory. The solution is used to compute the number of *Stops*. Figure 3 illustrates the result of the trajectory estimation and Stop detection algorithm.

Remark 1 (Data compression). *In our dataset[3], the average number of GPS points sent by a vehicle on a link is 9.6. The algorithm reduces the GPS trace to: entrance time in the link, travel time, number of stops. The amount of data to be processed by the subsequent algorithms is reduced by a factor of almost 10, which is crucial for large scale applications which process large amounts of historical data.*

Remark 2 (Fixing λ). *In (2), λ acts as a trade-off between the sparsity of the solution and the fit to the observations. Cross-validation is not appropriate in our setting to fix this parameter as it would require one to decimate the trajectory and use some observations for the learning and others for the validation. Instead, λ is chosen by computing the* Bayesian Information Criterion *(BIC), using [19] to estimate the number of degrees of freedom. A LARS implementation [4] allows efficient computation to choose λ and compute the corresponding solution.*

[2]The assumption is further justified by traffic modeling [7] which commonly assumes that each *Go* phase has constant speed

[3]The number of GPS points per link depends on the level of congestion (vehicles spend a longer amount of time on each link), average length of the links of the network and sampling frequency of the probe vehicles.

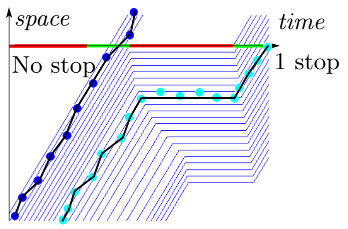

Figure 3: Example of trajectory estimation

3. TRAVEL TIME MODEL

We develop a travel time model which exploits the compressed data returned by the *Stop-&-Go* filter (number of stops and travel time experienced per link). The model captures the state transition probabilities: the probability of the number of stop on a link given the number of stops on the previous link of the trajectory. It also models the correlations of travel times for neighboring links given their state (number of stops). The travel time model is a combination of two models:

• A directed Markov model of discrete state random variables gives the joint probability distribution of the link states $\mathbb{P}_\theta(\{S_l\})$.

• A Gaussian Markov random field gives the joint distribution of the travel times $\mathbb{P}_\theta(\{Y_{l,s}\}, l \in \mathcal{L}, s \in \{0,\dots,m-1\})$.

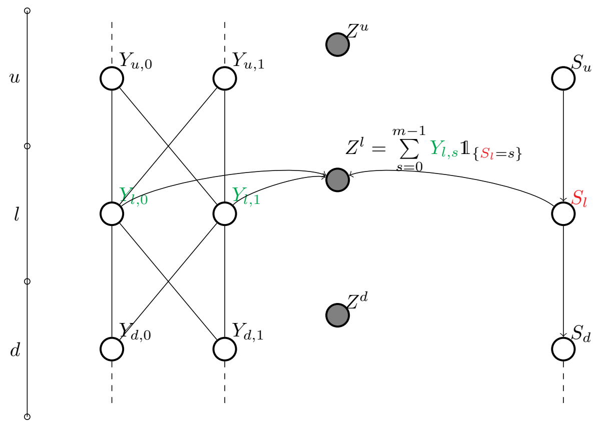

Figure 4: Graphical model of the dependency of link travel time Z^l on the state variable (e.g. number of stops) S_l and the conditional travel times $Y_{l,s}$. Here we consider the subgraph consisting of a link l, upstream link u and downstream link d on a given path.

Figure 4 presents the graphical model that encodes the dependencies between the link travel time Z^l, the link states S_l and the conditional travel times $Y_{l,s}$. The total travel time experienced by a vehicle on link l, is $Z^l = \sum_{s=0}^{m-1} Y_{l,s}\mathbb{1}_{\{S_l=s\}}$. The left portion of the figure shows a subset of the GMRF of travel time variables, and the right portion shows a subset of the Markov chain of states.

The graphical model shows that conditioning on the states experienced along a path allows one to compute the path travel time by summing over the corresponding variables in the GMRF. Further, when one conditions on the link travel times Z^l, then the two models become independent, which allows one to learn the models parameters separately. The rest of this section details the modeling and learning of the two models.

3.1 Markov model for state transitions

We consider the state variables $\{S_l\}_{l,\in\mathcal{L}}$. Each variable S_l has support $\{0,\dots,m-1\}$.

Given the path $p=(l_0,\dots,l_M)$ of a vehicle, the variables $\{S_{l_i}\}_{i\in\{1,\dots,M\}}$ have a Markov property, i.e. given the state of the *upstream link* l_{i-1}, the conditional state $(S_{l_i}|S_{l_{i-1}})$ is independent of the state of other upstream links:

$$\mathbb{P}(S_{l_i}|\{S_{l_{i-1}},\dots,S_{l_0}\})=\mathbb{P}(S_{l_i}|S_{l_{i-1}}) \tag{3}$$

We parametrize the model using an initial probability vector

$$\pi_s^l=\mathbb{P}(S_l=s) \tag{4}$$

and a transition probability matrix

$$T_{s_u,s_l}^{u\to l}=\mathbb{P}(S_l=s_l|S_u=s_u) \tag{5}$$

here $T_{s_u,s_l}^{u\to l}$ is the probability that l is in state s_l given that the upstream link u is in state s_u.

We learn the initial probability vector π^l and the transition probability matrices $T^{u\to l}$ of the Markov Chain by maximizing the likelihood of observing $(s^{(n)},y^{(n)})$. The log-likelihood is given by

$$\sum_{n=1}^{M}\left(\log(\pi_{s_0^{(n)}}^{l_0^{(n)}})+\sum_{u\overset{p^{(n)}}{\to}l}\log(T_{s_u^{(n)},s_l^{(n)}}^{u\to l})\right)$$

The parameters that maximize the log-likelihood are:

$$\pi_s^l\propto\bar{\pi}_s^l \quad\text{and}\quad T_{s_u,s_l}^{u\to l}\propto\bar{T}_{s_u,s_l}^{u\to l}$$

where $\bar{\pi}_s^l$ are the empirical counts of initial states and $\bar{T}_{s_u,s_l}^{u\to l}$ are the empirical transition counts. This solution corresponds to transitions and initial probabilities that are consistent with the empirical counts of initial states and transitions.

3.2 GMRF model for travel time distributions

We now present the model that describes the correlations between the travel time variables. We assume that the random variables $(Y_{l,s})_{l\in\mathcal{L},s\in\mathcal{S}_l}$ are Gaussian[4], and they can be stacked into one multivariate Gaussian variable $Y\sim\mathcal{N}(\mu,\Sigma)$ of size $m\,\mathrm{card}(\mathcal{L})$, mean μ and covariance Σ. Recall that $Y_{l,s}$ is the travel time on link l conditioned on state s, where $s\in\{0,\dots,m-1\}$. This travel time is a Gaussian random variable with mean $\mu_{l,s}$ and variance $\sigma_{l,s}$.

From the factorization property given by the Hammersley-Clifford Theorem, it is well known that the *precision matrix* $S=\Sigma^{-1}$ encodes the *conditional dependencies* between the variables. Since a link is assumed to be conditionally correlated only with its neighbors, the precision matrix is very sparse. Furthermore, this sparsity pattern is of particular interest: its pattern is that of a graph which is nearly planar. We take advantage of this structure to devise efficient algorithms that (1) estimate the precision matrix given some observations, and (2), infer the covariance between any pair of variables.

[4]While Gaussian travel times can theoretically predict negative travel time, in practice, these probabilities are virtually zero, as validated in Section 5.

As mentioned earlier, we have a set of N trajectories obtained from GPS data. After map-matching and trajectory reconstruction (Section 2), the set of observed trajectories $\{W_p : p=1,\cdots,N\}$ are sequences of observed states and variables (travel time):

$$\begin{aligned}W_p &= (l_1,s_1)\;(l_2,s_2)\;\cdots\left(l_{M_p},s_{M_p}\right)\\ y^p &= \left(y_{l,s}^p\right)_{(l,s)\in W_p}\end{aligned}$$

In our notations, y^p is an observation of the random vector $Y_p=(Y_i)_{i\in W_p}$, which is a M_p-dimensional marginal (or subset) of the full distribution Y. Hence Y_p is also a multivariate Gaussian with mean $\mu_{(W_p)}$ and covariance $\Sigma_{(W_p)}$ obtained by dropping the irrelevant variables (the variables that one wants to marginalize out) from the mean vector μ and the covariance matrix Σ. Its likelihood is thus the likelihood under the marginal distribution:

$$\begin{aligned}&\log p\left(y^p;\mu_{(W_p)},\Sigma_{(W_p)}\right)=\\ &C-\tfrac{1}{2}\log\left|\Sigma_{(W_p)}\right|-\tfrac{1}{2}\left(y^p-\mu_{(W_p)}\right)^T\Sigma_{(W_p)}^{-1}\left(y^p-\mu_{(W_p)}\right),\end{aligned} \tag{6}$$

where C is a constant which does not depend on the parameters of the model.

The problem of estimating the parameters of the model $\theta=(\mu,\Sigma)$ from the i.i.d. set of observations $\mathcal{D}=\{W_p,y^p : p=1,\cdots,N\}$ consists in finding the set of parameters $\theta^\star$ that maximize the sum of the likelihoods of each of the observations :

$$l(\theta|\mathcal{D})=\sum_{p=1}^{N}\log p\left(y^p;\mu_{(W_p)},\Sigma_{(W_p)}\right) \tag{7}$$

Unfortunately, the problem of maximizing (7) is in general not convex and may have multiple local minima since we have only partially observed variables y^p. A popular strategy in this case is to *complete* the vector (by computing the most likely completion given the observed variables). This algorithm is called the *Expectation-Maximization* (EM) procedure. In our case, the EM procedure is not a good fit for two reasons:

- Since we observe only a small fraction of the values of each vector, the vast majority of the values we would use for learning would be sampled values, which would make the convergence rate dramatically slow.
- The data completion step would create a complete $n-$size sample for each of our observation, thus our complexity for the data completion step would be $\mathcal{O}(Nn)$, which is too large to be practical.

Instead, we solve a related problem by computing sufficient statistics from all the observations. Consider the simpler scenario in which all data has been observed, and denote the empirical covariance matrix by $\tilde{\Sigma}$. The maximum likelihood problem to find the most likely precision matrix is then equivalent to:

$$\underset{S}{\textbf{minimize}}\;-\log|S|+\mathrm{Tr}\left(S\tilde{\Sigma}\right) \tag{8}$$

under the structured sparsity constraints $S_{uv}=0\;\forall\;(u,v)\notin\mathcal{E}$. The objective is not defined when S is not positive definite, so the constraint that S is positive definite is implicit. A key point to notice is that the objective only depends on a restricted subset of terms of the covariance matrix:

$$\mathrm{Tr}\left(S\tilde{\Sigma}\right)=\sum_{(u,v)\in\mathcal{E}}S_{uv}\tilde{\Sigma}_{uv}$$

This observation motivates the following approach: instead of considering the individual likelihoods of each observation individually, we consider the covariance that would be produced if all the observations were aggregated into a single covariance matrix. This approach discards some information, for example the fact that some variables are more often seen than others. However, it lets us solve the full covariance Problem (8) using partial observations. Indeed, all we need to do is estimate the values of the coefficients $\tilde{\Sigma}_{uv}$ for (u, v) a non-zero in the precision matrix. We present one way to estimate these coefficients.

Let N_i be the number of observations of the variable Y_i: $N_i = \text{card}(\{p : i \in W_p\})$. Combining all the observations that come across Y_i, we can approximate the mean of any function $f(Y_i)$ by some empirical mean, using the N_i samples:

$$\mathbb{E}_i [f(Y_i)] = \frac{1}{N_i} \sum_{p:i \in W_p} f(y_i^p) \tag{9}$$

Similarly, defining the number of observations of both Y_i and Y_j: $N_{i\cap j} = \text{card}(\{p : i \in W_p \text{ and } j \in W_p\})$, we can approximate the mean of any function $f(Y_i, Y_j)$ of Y_i, Y_j, using the set of observations that span both variables Y_i and Y_j:

$$\mathbb{E}_{i\cap j} [f(Y_i, Y_j)] = \frac{1}{N_{i\cap j}} \sum_{p\,:\,i,j \in W_p} f\left(y_i^p, y_j^p\right) \tag{10}$$

Using this notation, the empirical mean is $\hat{\mu}_i = \mathbb{E}_i [Y_i]$. Call $\hat{\Sigma}$ the *partial empirical covariance matrix* (PECM):

$$\hat{\Sigma}_{ij} = \begin{cases} \mathbb{E}_{i\cap j} [Y_i Y_j] - \mathbb{E}_i [Y_i] \mathbb{E}_j [Y_j] & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

Using this PECM as a proxy for the real covariance matrix, one can then estimate the most likely GMRF by solving the following problem:

$$\underset{S}{\textbf{minimize}} \; -\log |S| + \left\langle S, \hat{\Sigma} \right\rangle \tag{11}$$

Note that this definition is asymptotically consistent: in the limit, when an infinite number of observations are gathered ($N_{ij} \to \infty$), the PECM will converge towards the true covariance: indeed $\hat{\Sigma}_{ij} \to \mathbb{E}[Y_i Y_j] - \mathbb{E}[Y_i] \mathbb{E}[Y_j]$ and for all S, $\left\langle S, \hat{\Sigma} \right\rangle \to \left\langle S, \mathbb{E}\left[YY^T\right] - \mathbb{E}[Y] \mathbb{E}[Y]^T \right\rangle$.

Unfortunately, the problem is not convex because $\hat{\Sigma}$ is not necessarily positive semi-definite (even if the limit is), since the variables are only partially observed. For instance, if we have a partially observed bivariate Gaussian variable X: $(10, 10)$, $(-10, -10)$, $(1, \star)$, $(-1, \star)$, $(\star, 1)$, $(\star, -1)$, the empirical covariance matrix $\hat{\Sigma}$ has diagonal entries $(51, 51)$ and off-diagonal entries $(100, 100)$. Its eigenvalues are $-49, 151$ hence it is not definite positive.

There is a number of ways to correct this. The simplest we found is to scale all the coefficients so that they have the same variance:

$$\hat{\Sigma}_{ij} = \begin{cases} \alpha_{ij} \mathbb{E}_{i\cap j} [Y_i Y_j] - \mathbb{E}_i [Y_i] \mathbb{E}_j [Y_j] & \text{if } N_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\alpha_{ij} = \sqrt{\frac{\mathbb{E}_i\left[Y_i^2\right] \mathbb{E}_j\left[Y_j^2\right]}{\mathbb{E}_{i\cap j}\left[Y_i^2\right] \mathbb{E}_{i\cap j}\left[Y_j^2\right]}}$$

This transformation has the advantage of being local and easy to compute. This is why it is completed by an increase of the diagonal coefficients by some factor of the identity matrix.

Another problem is due to the relative imbalance between the distributions of samples: cars travel much more on some roads than others. This means that some edges may be much better estimated than some others, but this confidence does not appear in the PECM. In practice, we found that *pruning* the edges with too few observations improved the results

4. INFERENCE

Given the model parameters, the inference task consists in computing the probability distribution $\mathbb{P}(Z^{(p)} \le t)$ of the total travel time along a fixed path $(p) = (p_0, \ldots, p_I)$. The path travel time (i.e. total travel time along the path) is given by

$$Z^{(p)} = \sum_{s \in \mathcal{S}} \mathbb{1}_{\{S_{(p)} = s\}} Z^{(p)}_{|s}$$

where $\mathcal{S} = \{1, \ldots, m\}^I$ is the set of possible path states, and $Z^{(p)}_{|s}$ is the path travel time given the path state s, and is given by $Z^{(p)}_{|s} = \sum_{l \in p} Y_{l, s_l} = e(p, s)^T Y$. Here, $e(p, s)$ is a binary vector that selects the appropriate entries in the vector of travel times Y (corresponding to path p with states s):

$$e(p, s)_{(l, s'_l)} = 1 \Leftrightarrow l \in p \text{ and } s'_l = s_l$$

This vector $e(p, s)$ is very sparse and has precisely I non-zero entries. Using the law of total probability we have:

$$\mathbb{P}\left(Z^{(p)}\right) = \sum_{s \in \mathcal{S}} \mathbb{P}(S = s)\, \mathbb{P}\left(Z^{(p)}_{|s}\right) \tag{12}$$

The variable $Z^{(p)}_{|s} = e(p, s)^T Y$ is a linear combination of the multivariate variable Y, and so is also normally distributed:

$$Z^{(p)}_{|s} \sim \mathcal{N}\left(e(p, s)^T \mu, e(p, s)^T \Sigma e(p, s)\right) \tag{13}$$

The marginal distribution of travel times along a path is a mixture of univariate Gaussian distributions. There is however two problems for this algorithm to be practical. *(i)* The mixture from Equation (12) contains a term for each possible combination of states, and has size m^I. Enumerating all the terms is impossible for moderately large lengths of paths. *(ii)* In order to compute the variance of each distribution of the mixture, one needs to estimate the covariance matrix Σ. However storing (or computing) the complete covariance matrix is prohibitively expensive with millions of variables.

We find tractable solutions for *(i)* by using a sampling method on the Markov model to choose a tractable number of states, and *(ii)* by using a random projection method to construct a low-rank approximation of the covariance matrix Σ. Note that the mean of the complete distribution can be computed exactly in $O(Im^2)$ time, using a dynamic programming algorithm. Thus, our model is also applicable to situations that do not require the full distribution. We do not present this algorithm further due to space limitations.

Gibbs sampling. The sequence of state variables $\{S_l\}_l$ for a given path form a Markov chain, with initial probability

π^0 and transition probability matrices $T^{i-1\to i}$, which are given by the trained Markov model in Section 3.1. We can sample from the Markov chain by first sampling s_0 from the categorical distribution with parameter π^0, then sequentially sample s_i from the conditional distribution of $S_i|S_{i-1}$.

Using this procedure, we generate K samples $\hat{s}^1, \dots, \hat{s}^K$ from $\mathcal{S}$, the set of possible states. The complete (exponential) distribution can be approximated with the empirical distribution:

$$\mathbb{P}\left(Z^{(p)}\right) = \sum_s \hat{w}_{p,s}\, \mathbb{P}\left(Z^{(p)}_{|s}\right) \tag{14}$$

in which each weight $\mathbb{P}$ is the fraction of samples corresponding to the state s:

$$\hat{w}_{p,s} = \frac{1}{K} \operatorname{card}\{k | \hat{s}^k = s\} \tag{15}$$

The sum (14) contains at most K terms, since at most K approximate weights are positive, and converges towards the true distribution of states as the number of samples increases. In practice, numerical experiments showed that it was only necessary to sample a relatively small number of states. Section 5 includes a numerical analysis and evaluation of the sampling method.

Low rank covariance approximation. Compute the variances of each sequence of variables:

$$e(s)^T \, \Sigma \, e(s)$$

with s being a valid sequence of variables in the graph of variables.

Using the full covariance matrix Σ to estimate the covariance of each path $\sigma(s)^2 = e(p,s)^T \Sigma e(p,s)$ is impractical for two reasons: as mentioned before, we cannot expect to compute and access the full covariance matrix, and also the sum $e(p,s)^T \Sigma e(p,s)$ sums I^2 elements from the covariance matrix. Since we do not need to know the variance terms with full precision, an approximation strategy using random projections is appropriate. More specifically, we use the following result from [1]:

Given some fixed vectors $v_1 \cdots v_n \in \mathbb{R}^d$ and $\epsilon > 0$, let $R \in \mathbb{R}^{k\times d}$ be a random matrix with random Bernoulli entries $\pm 1/\sqrt{k}$ and with $k \geq 24\epsilon^{-2} \log n$. Then with probability at least $1 - 1/n$:

$$(1-\epsilon)\|v_i\|^2 \leq \|Rv_i\|^2 \leq (1+\epsilon)\|v_i\|^2$$

Call R such a matrix. Consider the Cholesky decomposition of the precision matrix, $S = LL^T$. Then

$$\begin{aligned} \sigma(s)^2 &= e(p,s)^T \Sigma e(p,s) \\ &= e(p,s)^T L^{-T} L^{-1} e(p,s) \\ &= \left\|L^{-1} e(p,s)\right\|^2 \end{aligned}$$

From the following lemma, we can approximate this norm:

$$\hat{\sigma}^2(s) = \left\|Q^T e(p,s)\right\|^2 = \left\|\sum_{i=1\cdots I} Q_{(l_i,s_i)}\right\|^2 \tag{16}$$

with $Q = L^{-1}R$ and R defined as obtained from the lemma above. Computing the approximate variance $\hat{\sigma}^2$ requires the addition of I vectors of size k. In practice, this summing operation is vectorized and very fast.

This method assumes we can efficiently compute the Cholesky factorization, and that inversion operation $L^{-1}x$ is efficient as well. In our case, the graph of the GMRF is nearly planar. Some very efficient algorithms exist that compute the Cholesky factorization in near-linear time ([2]). In practice, computing the Q matrix is very fast.

5. EVALUATION

The article presents an algorithm to turn GPS traces from a small fraction of the vehicles traveling on the road network into valuable traffic information to develop large scale traffic information platforms and optimal and robust routing capabilities. The value of the model depends crucially on the quality of the estimates of point to point travel time distribution. Another key feature of the algorithm is its computational complexity and its ability to scale on large road networks. The large number of road segments and intersections leads to high-dimensional problems for any network of reasonable size. Moreover, the estimate for travel time distributions between any two points on the network needs to be computed in real-time. It would not be acceptable to wait more than a few seconds to get the results. This section first analyzes the quality of the learning and the estimation of the model. Then, we study the computational complexity of the algorithm and its ability to scale on large networks. In particular, some aspects of the algorithm arise as tradeoffs between the computational complexity and the desired precision in the learning and real-time inference.

The validation results are based on anonymized GPS traces provided by a GPS data aggregator. We consider an arterial network in the Bay Area of San Francisco, CA with 506,585 links. The algorithm processes 426 million GPS points, aggregated from 2,640,319 individual trajectories. Each trajectory is less than 20 minutes long for privacy reasons.

5.1 Travel time distribution

The full validation of the performance of the algorithm requires the observation of the travel time of every vehicle on every link of the network. This mode of validation is unfortunately not available for any reasonably sized network. We validate the learning capabilities of the algorithm using data which was not used to train the Markov model GMRF. On this validation dataset, we perform two types of validation: (i) a path-level validation (a limited set of individual paths are evaluated) and (ii) a network-level validation (metrics taken over the entire validation dataset).

Comparison models

In this Section, the results of the model are compared to simpler models which arise as special cases of the model presented in the article. We introduce these models and denominations, which we use throughout the evaluation.

- *One mode independent*: The travel time distribution on each segment is Gaussian. The travel time on distinct segments are independent random variables.
- *One mode*: The travel time distribution on the network is a multi-variate Gaussian (one dimension per link). In the precision matrix, element (i,j) may be non-zero if i and j map to neighboring links in the road network.
- *Multi-modal independent*: Same as the MM-GMRF, excepted that the covariance matrix of the multi-variate Gaussian is diagonal, imposing that given the mode, the travel times on different links of the network are independent.

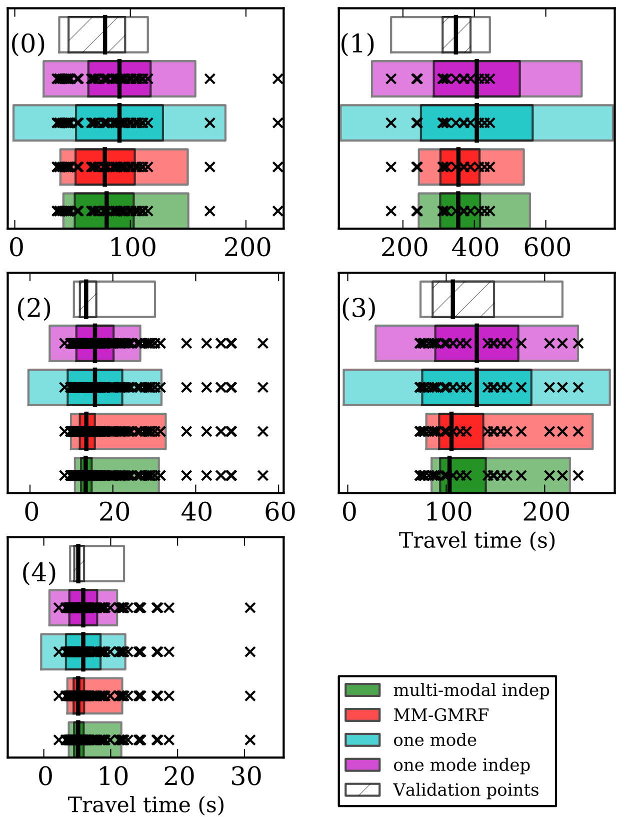

Figure 5: 50% and 90% confidence intervals computed by the different models and compared to the validation travel times on the selected paths.

The model developed in this article is referred to as the *MM-GMRF* model.

Path validation

Most probe vehicles have different paths throughout the road network. Among the trajectories of the vehicles in the validation set, we select a set of paths for which a large enough number of vehicles has traveled to perform statistical validation of the distribution of travel times. We impose a minimum length (set to 150 m in the numerical experiments) on these paths to ensure that we validate the learning of the spatial distributions of the modes (Section 3.1) and the spatial correlations between each mode (Section 3.2). The paths are selected for having the largest amount of validation data.

For each selected path, the box plots on Figure 5 compare the 50% and 90% confidence intervals of the validation data collected on the path (top box) with the intervals computed by the different models (*Multi-modal independent*, *MM-GMRF* (our model), *One mode* and *One mode independent*, from top to bottom). We also display the median travel times as a vertical black line, both for the validation data and the different models. Scatter crosses, representing the validation travel times, are super-imposed to the results of each model to improve visualization.

We notice a significant difference in the results between the uni-modal models (*One mode* and *One mode independent*) and the multi-modal models (*Multi-modal independent*, *MM-GMRF*). The uni-modal models tend to over-estimate both the median and the variance of travel times. These models cannot account for the difference of travel times due to stops on trajectories, which is one of the main features of arterial traffic [7]. The over-estimated variance illustrates why it is important to incorporate the variability of travel times due to stops in the structure of the model. On the other side, the multi-modal models are able to capture the features of the distribution fairly accurately. The differences in accuracy between the *Multi-modal independent* model and the MM-GMRF model (which takes into account correlations in the Gaussian distribution) are not significant, even though the model with correlations estimates the variability slightly more accurately. It seems that capturing the variability of travel times due to stop is the most important feature of the model.

In Figure 6, we display the cumulative distribution of the validation data and the cumulative distribution of each model for the same paths as for Figure 5 (displayed in the same order). The figure displays more precisely the difference in the estimation accuracy of the different models. As seen in Figure 5, the multi-modal models are more accurate than their uni-modal counterparts.

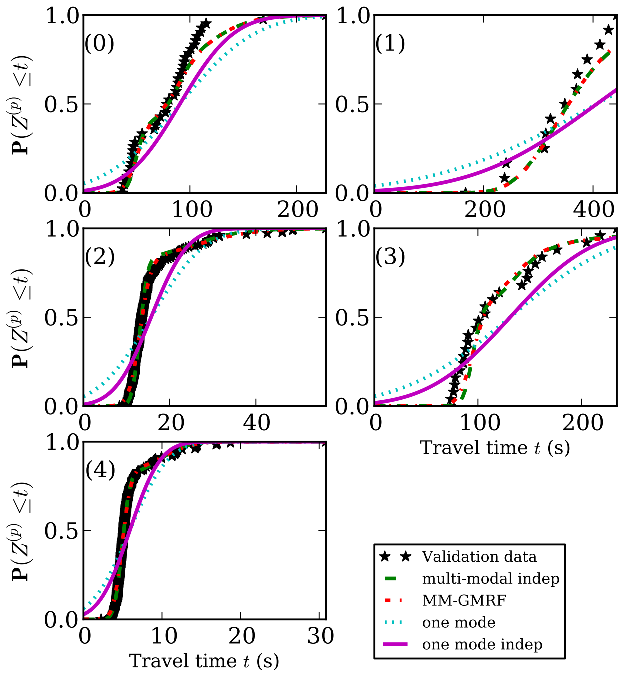

Figure 6: Cumulative distribution of travel times computed by the different models and compared to the validation travel times received on the selected paths.

Network scale validation

Most points in the observation dataset represent different paths for the probe vehicles. For this reason, the distributions cannot be compared directly. Instead, we compute the log-likelihood of each validation path and analyze the quality of the travel time bounds provided by the distribution for each path.

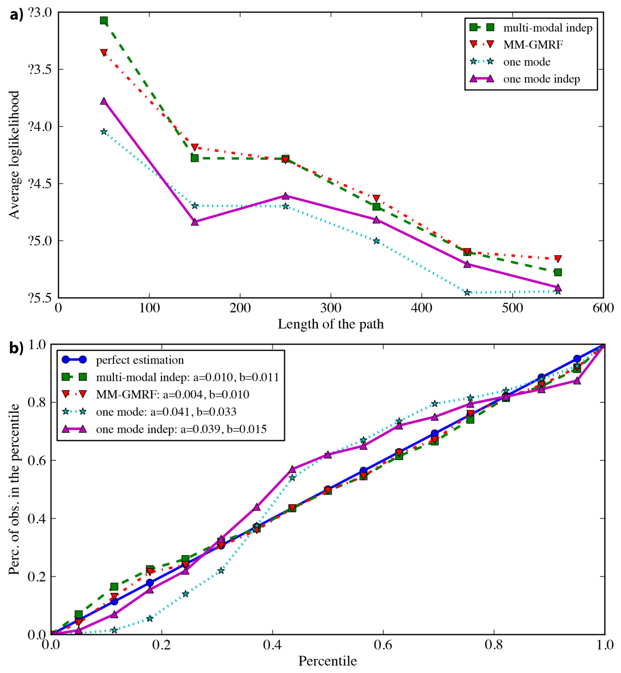

Figure 7: a) Average log-likelihood of the validation paths (by path length), for each model. b) Validation of the distribution percentiles for each model.

Figure 7 a) displays the average likelihood of the validation paths computed by the different models. The figure also analyzes how the path length influences the results. There are two motivations for doing so: (i) the length of the path influences the support of the distribution (longer paths are expected to have a larger support) which may affect the likelihood and (ii) the different models may perform differently on different lengths as they do not take into account spatial dependencies in a similar way.

As was expected from the analysis of Figures 5 and 6, the multi-modal models perform better than their uni-modal counterparts. Compared to the path validation results, the figure shows more significantly the effect of correlations. The figure shows slight improvements for the multi-modal model which takes into account the correlations. Surprisingly, the contrary is true for the uni-modal models. We also notice that the likelihood decreases with the length of the path, as we were expecting.

Figure 7 b) analyzes the quality of the travel time distribution computed on the network. For that, we use a *p-p plot* (or percentile-percentile plot) which assesses how much each learned distribution matches the validation data. To each path p in the validation dataset corresponds an inverse cumulative distribution $\mathbb{P}_p^{-1}$ (computed from the trained model) and a travel time observation z^p. A point (α, β) on the curve corresponds to having β percent of the validation points such that $z^p \leq P_p^{-1}(\alpha)$. If the estimation was perfect, there would be exactly $\alpha\%$ of the data points in the percentile α. To quantify how much each model deviates from perfect estimation, we display two metrics denoted a (above) and b (below). Let f correspond to the p-p curve of a model, the corresponding metrics are computed as follows:

$$a = \int_0^1 \max(f(\alpha) - \alpha, 0)\, d\alpha\,,\; b = \int_0^1 \max(\alpha - f(\alpha), 0)\, d\alpha$$

These values provide insight on the quality of the fit of the model. For example, a model with a large a-value tends to overestimate travel times. Similarly, a model with a large b-value tends to underestimate travel times. Both uni-modal models have large a-values. The large variance estimated by the models (already noticed in Figure 5) to account for the variability of travel times leads to non-negligible probability densities for small travel times which are not physically possible. Compared to the likelihood validation of Figure 7 a), the p-p plot analyzes the quality of fit for different percentiles of the distribution. In particular, we notice that the effect of capturing the correlations in the multi-variate model mostly affects the estimation of the low and high percentiles in the distribution. We expect that this is due to the fact that correlations accounts for the impacts of slow vs. fast drivers or congested vs. lest congested conditions.

5.2 Sampling

In this section, we discuss numerical results regarding the quality of approximate inference using the Gibbs sampling method, on a fixed path on the network.

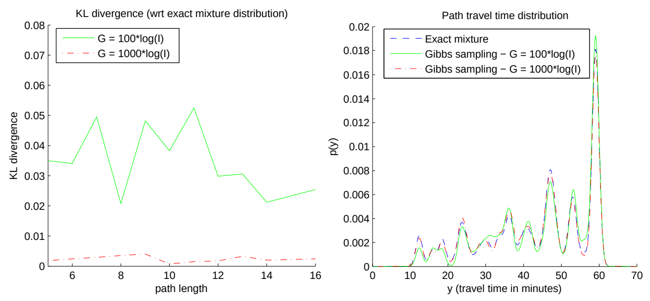

Figure 8: KL-divergence between the approximate distribution and the exact mixture distribution, as a function of the path length (left), and example distributions for a path of length $I = 17$ (right).

Figure 8 shows the Kullback-Leibler divergence of the approximate distributions, with respect to the exact distribution. We compare two runs of the sampling, with respective sizes $G = 100\log(I)$ and $G = 1000\log(I)$ where I is the length of the path in links. The divergence measures the similarity between two distributions. As can be seen, even a small number of samples (relative to the total size of the mixture) leads to a very close approximation.

5.3 Scaling

In this section, we discuss the scalability aspects of the learning algorithm (it is clear from the discussion in Section 3.2 that the inference is independent from the size of the network). We ran the learning for networks defined by different bounding boxes. The bounding boxes were adjusted so that the number of links in each subnetworks had different orders of magnitude. The longest step by far is the training of the GMRF. We report in Figure 9 the training time for different networks, all other parameters being equal. As one can see, the training time increases linearly with the number

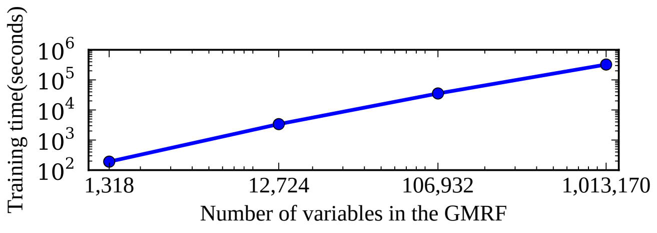

Figure 9: Log-log plot of the training time, as a function of the size of the GMRF.

of variables of the GMRF over a large range of network sizes. The graphs associated to each GMRF are extremely sparse: the average vertex degree of the largest graph is 9.46.

6. CONCLUSIONS

The state of the art for travel time estimation has focused on either precise and computationally intensive physical models, or large scale, data-driven approaches. We have presented a novel algorithm for travel time estimation that aims at combining the best of both worlds, by combining physical insights with some scalable algorithms. We model the variability of travel times due to stops at intersections using a *Stop-&-Go* filter (to detect stops) and a Markov model to learn the spatial dependencies between stop locations. We also take into account the spatio-temporal correlations of travel times due to driving behavior or congestion, using a Gaussian Markov Random Fields. In particular, we present a highly scalable algorithm to train and perform inference on Gaussian Markov Random Fields, when applied on geographs.

We analyze the accuracy of the model using probe vehicle data collected over the Bay Area of San Francisco, CA. The results underline the importance to take into account the multi-modality of travel times in arterial networks due to the presence of traffic signals. The quality of the results we obtain are competitive with the state of the state of the art in traffic, and also highlight the good scalability of our algorithm.

7. REFERENCES

[1] D. Achlioptas. Database-friendly random projections. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 274–281. ACM, 2001.

[2] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam. Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software (TOMS)*, 35(3):22, 2008.

[3] J. Duchi, S. Gould, and D. Koller. Projected subgradient methods for learning sparse gaussians. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, 2008.

[4] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[5] L. Gu, E. P. Xing, and T. Kanade. Learning gmrf structures for spatial priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*, pages 1–6. IEEE, 2007.

[6] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen. Learning the dynamics of arterial traffic from probe data using a dynamic bayesian network. *IEEE Transactions on Intelligent Transportation Systems,*, 13(4):1679 –1693, Dec. 2012.

[7] A. Hofleitner, R. Herring, and A. Bayen. Arterial travel time forecast with streaming data: a hybrid flow model - machine learning approach. *Transportation Research Part B*, Nov. 2012.

[8] T. Hunter, P. Abbeel, and A. M. Bayen. The path inference filter: Model-based low-latency map matching of probe vehicle data. In *Algorithmic Foundations of Robotics X*, Springer Tracts in Advanced Robotics, pages 591:1–607:17, Heidelberg, Germany, 2012. Springer-Verlag.

[9] M. I. Jordan, editor. *Learning in graphical models.* MIT Press, Cambridge, MA, USA, 1999.

[10] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[11] S. Lim, C. Sommer, E. Nikolova, and D. Rus. Practical route planning under delay uncertainty: Stochastic shortest path queries. *In RSS - Robotics: Science and Systems VIII*, 2012.

[12] D. M. Malioutov, J. K. Johnson, M. J. Choi, and A. S. Willsky. Low-rank variance approximation in gmrf models: Single and multiscale approaches. *IEEE Transactions on Signal Processing*, 56(10):4621–4634, 2008.

[13] M. Ramezani and N. Geroliminis. On the estimation of arterial route travel time distribution with markov chains. *Transportation Research Part B*, 46(10):1576–1590, 2012.

[14] S. Samaranayake, S. Blandin, and A. Bayen. A tractable class of algorithms for reliable routing in stochastic networks. *Transportation Research Part C*, 20(1):199–217, 2011.

[15] R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[16] D. Tiesyte and C. S. Jensen. Similarity-based prediction of travel times for vehicles traveling on known routes. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems.* ACM, 2008.

[17] B. S. Westgate, D. B. Woodard, D. S. Matteson, and S. G. Henderson. Travel time estimation for ambulances using bayesian data augmentation. *Submitted, Journal of the American Statistical Association*, 2011.

[18] F. Zheng and H. van Zuylen. Reconstruction of delay distribution at signalized intersections based on traffic measurements. In *Proceedings of 13th IEEE Intelligent Transportation Systems Conference*, pages 1819–1824, Sept. 2010.

[19] H. Zou, T. Hastie, and R. Tibshirani. On the "degrees of freedom" of the lasso. *The Annals of Statistics*, 35(5):2173–2192, 2007.