Probabilistic estimation of link travel times in dynamic road networks

Mohammad Asghari*, Tobias Emrich*, Uğur Demiryurek, Cyrus Shahabi

*Integrated Media Systems Center, University of Southern California
{masghari, emrich, demiryurek, shahabi}@usc.edu

ABSTRACT

Due to the availability of large historical and real-time traffic data, car navigation systems are becoming more and more advanced in predicting the travel time for various routes and finding the fastest route from a source to a destination given a start time. The most advanced of these systems predict the travel time of the routes, given past traffic patterns in order to find the best route. However, the best route is not necessarily a reliable route as well, i.e., the route with the least variation in possible travel times. The most reliable route is desirable when traveling with a deadline, e.g., to reach a flight at the airport or to arrive on time for an important meeting. To find the most reliable route, one needs to predict the probability distribution of travel times for that route. This in turn requires the estimation of travel time probability distributions for each and every link, given a link-entrance-time. In this paper we address the problem of computing these link travel time distributions. To the best of our knowledge there has not been any study on how to compute probability distributions for links (edges) in road networks. We show how this first step can affect the accuracy of the travel time distribution over the entire route. Our final challenge is to evaluate the result of different approaches in computing these travel time distributions, which is difficult because the reported travel time is not a single value but a probabilistic distribution highly depending on the trip start time. We thus propose a statistical test that enables us to evaluate these outcomes.

Categories and Subject Descriptors
H.2.8 [Database Management]: Database Applications-Spatial databases and GIS; I.3.4 [Information Storage and Retrieval]: Systems and Software-Question-answering (fact retrieval) systems

Keywords
Probabilistic travel times, route reliability, dynamic road networks

1. INTRODUCTION

Navigation systems have changed the way we drive and in particular affected how we plan our trips. Nowadays, they are not only used to find the shortest route in an unknown area but also to react to current events such as traffic jams and accidents and compute the fastest route. The latter became possible due to the availability of real-time traffic data gathered from road sensors, GPS transmissions from vehicles and reporting tools for accidents. The most recent navigation systems (e.g., [1]) even go one step further and use the collected historic data to build traffic models in order to predict the future traffic. In conjunction with the real-time information, the ultimate goal is to provide a faster route and better estimation of the expected travel time to the user. Since a prediction model is involved in these systems, a main observation to be made is that the outcome, i.e., the estimated time for a route and thus the fastest route, can never be trusted with certainty and is subject to variation. Consequently, the fastest route is not necessarily the most reliable route, i.e., the route with the least variation in possible travel times. Reliability of a route is particularly important to users when a deadline, such as a flight departure or an important meeting, has to be met.

To illustrate, consider the example in Fig. 1(a), where estimated total travel times for two routes with the same source and destination are given. Based on the expected travel times, route A is the better choice. However, this result ignores the uncertain nature of traffic prediction. On the other hand, in Fig. 1(b), travel times for the same two routes are given with some level of uncertainty, i.e., the travel times are given by a probability density function (pdf). Continuing with our example in Fig. 1, let us assume that the user has only 60 minutes to reach the destination in order to be on time. The probability that the user is at the destination in at most 60 minutes, is the sum of probabilities of travel times smaller or equal to 60 minutes. In the example in Fig. 1(b), the user has a chance of 99.2% to reach his destination on time when choosing route B, whereas choosing route A gives him only an 89% chance to be on time. Furthermore, if the user wants to achieve the same level of confidence using route A, he has to start 10 minutes earlier.

The example in Fig. 1 shows that similar to other applications [2], explicitly handling the uncertainty is better than ignoring it. Correspondingly, we believe that taking the uncertainty of the travel time into account will enable next-generation route planning applications by extending path queries to probabilistic path queries, al-
lowing for queries such as: *What are the paths from my hotel to the airport whose travel times are at most 40 minutes with a probability of at least 95%?*

To achieve this goal, we have to compute a *pdf* for the travel times over a given route which gives the probability of each possible travel time, being the actual travel time of the route. Many solutions have been proposed to compute such *pdfs* [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. A common assumption in all these approaches is that the probabilistic link travel times (*pltts*), i.e., the probabilistic times to travel each link on the route, is given a priori. In other words, these studies focus on computing a *pdf* for the entire route by combining the *pdfs* of all links in different ways. On the other hand, to the best of our knowledge, no study has focused on finding the *pltts* for a single link. All existing techniques on predicting link travel times only provide a single estimated value for each link [1, 15] which is not compatible with the algorithms developed to compute the *pdf* of a route. In this paper, we aim to fill in this gap by providing several techniques to compute *pltts* in a road network.

Even though it is possible to evaluate these techniques by comparing how accurate they predict link travel times, the overall goal is yet to build a *pdf* that predicts the travel time over a route as accurate as possible. For this reason, we utilize our *pltts* computation techniques in different algorithms proposed for computing route *pdfs*. In order to cover all different algorithms, we first classify them based on three basic characteristics: representation, time-dependency and correlation. The significance of the *pltts* computation step becomes clear once we show that the accuracy of the *pltts* directly impact the accuracy of the *pdf* for the entire route. In fact, through experiments on real-world data, we show that utilizing a good algorithm for generating route *pdfs* with a bad *pltts* estimation technique, will result in a less accurate outcome compared to when a bad algorithm is utilized with a good *pltts* estimation technique.

As mentioned earlier, no existing study has focused on *pltts* computation. Consequently, no study has been able to perform an end-to-end evaluation of different algorithms for computing route *pdfs*. For the first time, in this paper, we use real-world data sets to evaluate these algorithms. Evaluating the accuracy of a probabilistic prediction is a daunting task itself. We show how several existing methods such as expected distance, goodness-of-fit and scoring functions are incapable of this task. Therefore, we propose a solution which is capable of evaluating the accuracy of both *pltts* and route *pdfs*.

In sum, the contributions of this paper are as follows:

- Propose techniques to compute *pltts* as a missing step towards computing a *pdf* for travel times of a route (Section 3).
- Classify existing algorithms for computing route *pdfs* based on their basic characteristics (Section 4).
- Propose a measure capable of evaluating the accuracy of a probabilistic prediction (Section 5).
- Conduct various experiments using real-world data to (1) compare different *pltts* computation techniques, (2) perform an end-to-end comparison of different algorithms for building route *pdfs* and more importantly (3) show the impact of *pltts* computation in the overall process of generating route *pdfs* (Section 6).

2. **PROBLEM DEFINITION**

In this section, we define the terminologies used in the paper and give a formal definition of the problem under consideration.

**Definition 1** *(Probabilistic Link Travel Times)*. In a road network, for every link *(i, j)* we define the probabilistic link travel time of link *(i, j)*, $c_{ij}(x)$, as the probability of taking $x$ seconds for a vehicle to traverse link *(i, j)* starting at time $t$.

The link travel times are thus time dependent and differ depending on the arrival time of the vehicle at that starting point of the link. Additionally we consider $c_{ij}$ to be a random variable with $f_{ij}$ representing its *pdf*.

**Definition 2** *(Route PDF (PMF))*: Assuming $p_{sd}$ represents a path starting from point $s$ and ending in point $d$, we define $\sigma_{sd}(x)$ as the random variable representing the travel time on $p_{sd}$ when we start at time $t$. Accordingly, the route *pdf*, $f_{sd}(x)$, gives the probability of taking $x$ seconds for a vehicle to traverse $p_{sd}$ starting at time $t$.

The problem considered in this paper is how to accurately estimate $c_{ij}$.

3. **PROBABILISTIC LINK TRAVEL TIMES**

The prediction of an event is usually tainted with uncertainty. Therefore, when predicting link travel times it is not possible to produce an accurate prediction, since there are too many influencing factors like road conditions, accidents, weather, time of the day, day of the week or season. The methods we propose, produce a probabilistic prediction which includes all the effects that have been observed before (historic data) and are influencing the traffic flow right now (current situation). For example if a street segment has often been the place of an accident (e.g., every other day between 9:00am and 10:00am) then the historical data will capture this effect and assign a high probability (e.g., 50%) to the prediction that the travel time on that segment will be higher than usual. In this section we will develop three techniques to estimate probabilistic link travel times and correlations between them. Before we continue, we need to introduce and explain a few concepts that will be used throughout this section. *Start time* $(t_s)$, is the time for which we want to predict the travel time. For example, if we want to know how long it will take to travel from point A to point B if we leave at 8:00am, then the *start time* will be 8:00am. *Query time* $(t_q)$, is the time at which we are making the prediction. For example, if the current time is 7:00am and we want to know how long it will take to travel from point A to point B if we leave at 8:00am, then the *query time* will be 7:00am. As it was seen in this example, *start time* and *query time* necessarily are not the same. We call the difference between the two the *prediction time* $(t_p)$.

**Probabilistic Link Travel Times**: The probabilistic link travel time of a link *(i, j)* at time $t$, $c_{ij}$, can be represented with two models; discrete or continuous.

To represent the link travel time $c_{ij}$ with a continuous *pdf*, an appropriate function is needed. Recent studies suggest that in road networks, the link travel times are normally distributed (e.g. [8]) or gamma distributed. [16] shows how to compute the sum of several gamma distributed random variables. The methods in [16] can be combined with the same approaches discussed here to compute route *pdfs* for routes consisting of links with gamma distributed travel times. In this study we only focus on link travel times that are normally distributed. In the case of a normal distribution, the random variable $c_{ij}$ is characterized by a mean $\mu_{ij}$ and a standard deviation $\sigma_{ij}$.

In the discrete case, the link travel time is represented by a discrete probability mass function. Therefore, the time domain has to be discretized. The simplest discretization scheme, known as b-discrete, divides the time domain $T = \{t | t = n \cdot \phi \land n \in \mathbb{N}\}$
evenly into intervals of length $\phi$. The corresponding probability mass function $F_{ij}$ of link travel time $c_{ij}$ reads:

$$F_{ij}(b) = \begin{cases} \int_{b}^{b+\phi} f_{ij}(w) dw & b = 0, \phi, \ldots, (L - 1)\phi \\ \int_{b}^{\infty} f_{ij}(w) dw & b = L\phi \\ 0 & \text{otherwise} \end{cases}$$

(1)

where $L\phi$ is the maximal considered time horizon in the future.

In the remainder of this section, we will discuss our techniques to generate probabilistic link travel-times.

3.1 Prediction through Historical Data

To obtain a pdf representing $c_{ij}^t$ for a start time ($t_s \geq t_0$) a simple approach is to use the available historical data. We define the set of historical data as $H = \{h|h < t_0\}$. Let us note that the link travel time $c_{ij}^h$, where $h < t_0$ is not a random variable but a certain value which is known. In order to predict the travel time for time $t_s \geq t_0$ more accurately, we only consider a subset of historical data, $H^{t_s} \subset H$, with similar characteristics to time $t_s$. For example, if we want to predict the behaviour of edge $(i, j)$ at 9:00am for the next Tuesday, an adequate set $H^{12}$ might consist of data points $h$, where $h$ is 9:00am on a Tuesday in the last year. The travel time $c_{ij}^h$ corresponding to last Friday 10:00pm might on the other hand, not be valuable in the prediction.

How to choose a set $H^{t_s}$ highly depends on the characteristics of the underlying traffic network. We use the results in [17] for choosing the appropriate $H^{t_s}$ for each start time. For a specific time of day during a weekday, $H^{t_s}$ consists of data points corresponding to the same time of day during any weekday. For example, if the start time is 9:00am on this Tuesday, $H^{12}$ will consist of data points at 9:00am during Mondays, Tuesdays, Wednesdays, Thursdays and Fridays. The assumption is that traffic patterns during weekdays are similar. More details can be found in [17].

For continuous representation of the link travel time, we assume normal distribution with the following parameters

$$\mu_{ij}^t = \frac{1}{|H^{t_s}|} \sum_{h \in H^{t_s}} c_{ij}^h$$

(2)

$$\sigma_{ij}^t = \frac{1}{|H^{t_s}|} \sum_{h \in H^{t_s}} (c_{ij}^h - \mu_{ij}^t)^2$$

(3)

$$\rho_{ij-kl} = \frac{\sum_{h \in H^{t_s}} (c_{ij}^h - \mu_{ij}^t)(c_{kl}^h - \mu_{kl}^t)}{|H^{t_s}| - 1}$$

(4)

where $\rho$ is the Pearson correlation coefficient [18].

For the case of discrete representation a pmf we set

$$F_{ij}^t(b) = \frac{1}{|H^{t_s}|} \sum_{h \in H^{t_s}} I([c_{ij}^h]^{\phi} = b)$$

(5)

where $[x]^{\phi}$ rounds $x$ up to the next multiple of $\phi$ and $I(c_{ij}^h = b)$ is an indicator variable which is 1 if $c_{ij}^h = b$ and 0 otherwise. The pmf of each edge is thus given by a histogram assigning for each possible travel time of the edge a probability corresponding to its proportional occurrence in the historical data.

Fig. 2 shows the discrete and continuous models of a typical inbound street segment of the network with a length of 1 mile for different times of the day. Based on the observation of historic patterns, in the morning a lot of traffic passes through this link. More traffic, intuitively, implies a higher probability of accidents which explains the rather large variation in the link travel time. At noon the traffic is reduced and in the evening the link travel time has the least amount of traffic, yielding the fastest travel time and in this case also the smallest variance. The figure also shows that the use of a normal distribution might not always adequately represent the link travel time.

We end this subsection with a note on query time and prediction time. When only historic data is used for prediction, the concept of prediction time becomes irrelevant. This means that if we are predicting for start time set to 9:00am today and we only use historic data, whether the query time is 7:00am or 8:00am does not make any difference since we only use $H^{t_s}$ corresponding to the start time to build the model. Prediction time and query time become important once we take the current situation into account when building the prediction model which we will discuss in the following.

3.2 Historical Data and Current Situation

The technique discussed in Section 3.1 provides link travel time distributions only based on historical data. These estimations may be a good choice if the start time $t_s$, for which the link travel time has to be predicted, is reasonably far (e.g., more than several hours) in the future. However, it may not be sufficient to capture current (and near future) traffic conditions with high confidence. For example, if we want to predict the travel time for 5 minutes later, the current situation of the road network might be more relevant than the historic data. Therefore, we present two estimation methods, incorporating both, the historical as well as the current link travel times of edges.

3.2.1 Prediction through Linear Interpolation

The intuition of this approach is that

- for a start time $t_s$ which is far in the future, the historical prediction (Section 3.1) is expected to yield a good prediction performance and
- for the start time $t_s = t_q$, the current situation yields the best "prediction".

Thus, for a start time $t_s$ which is in the relatively near future, we argue that both, the historical as well as the current situation should influence the prediction. The closer the time $t_s$ is to $t_q$ (smaller prediction time) the more weight the current situation should have and vice versa. As the prediction time increases, we eventually get to a point where the current situation should loose it’s influence. Towards this end we define a threshold parameter $\tau$, which defines the time-horizon in which the current situation has an influence on the prediction. Using a small time horizon $\tau$ is basically valuing the historic data, whereas setting $\tau$ to a larger value puts more weight on the current situation. We also define $\theta \in [0, 1]$ as the relative importance of the current situation in the estimation. $\theta$ depends on the prediction time and is computed as:

$$\theta = \min\left(\frac{t_p}{\tau}, 1\right)$$

(6)

where, as explained earlier, $t_p$ is the prediction time.

For continuous representation of link travel times, we estimate the parameters of the normal distribution as:

$$\mu_{ij}^t = \left(\theta \cdot \frac{1}{|H^{t_s}|} \sum_{h \in H^{t_s}} c_{ij}^h\right) + (1 - \theta) \cdot c_{ij}^q$$

(7)

$$\sigma_{ij}^t = \sqrt{\theta^2 \cdot \frac{1}{|H^{t_s}|} \sum_{h \in H^{t_s}} (c_{ij}^h - \mu_{ij}^t)^2}$$

(8)

In particular, we perform a linear interpolation between the current link travel time, and the summarized historical link travel time.
in order to find the link travel time to be estimated. Fig. 3 illustrates this concept, where \( \theta = 0.5 \) and thus the expected value of the prediction is the average of the current situation and the predicted value by the historical approach. Accordingly, the standard deviation of the historical prediction is cut by half.

\[
F_{i,j}^{\lambda}(b) = \frac{1}{|H^{\pi}|} \sum_{h \in H^{\pi}} I([\theta \cdot c_{i,j}^{h} + (1 - \theta) \cdot c_{i,j}^{s}]^\phi = b) \quad (9)
\]

where \([x]^\phi\) rounds \(x\) up to the next multiple of \(\phi\) and \(I()\) is the indicator function.

### 3.2.2 Prediction through Similar Historical Data

The third and final approach we propose is based on the idea that the current situation is a crucial indicator of how the travel time of a link will develop in the future. In this approach we use the current situation to further restrict the set \(H^\pi\). Specifically we only include times \(h \in H^\pi\), for which link \((i,j)\) had a travel time similar to the current situation at time \(h - t_p\). In this context, we define similarity by a percentage threshold \(\lambda\), which represents the deviation from the current travel time. To illustrate this idea, we provide an example. Assume we are predicting the travel time for link \(l\) where \(t_0 = 7:00am\) and \(t_s = 8:00am\). We only consider the data points at 8:00am from days where the traffic at 7:00am was similar to today. For example, if the current travel time over link \(l\), is 5 minutes and \(\lambda = 0.2\), we only consider days where the travel time over link \(l\) is somewhere between 4 minutes and 6 minutes.

We can formally define the further restricted set \(H_{i,j}^\pi\) as:

\[
H_{i,j}^\pi = \{h | h \in H^\pi \wedge \left| c_{i,j}^{h} - c_{i,j}^{l_p} \right| \leq \lambda \} \quad (10)
\]

Figure 2: Historical model for a street segment for different times of a Monday

Figure 3: Linear interpolation of ltt

In the case of discrete link travel times, we adapt the same concept and obtain the following parameters.

\[
\lambda = \frac{5 \text{ minutes}}{2} \approx 2.5 \text{ minutes}
\]

The value of \(\lambda\) affects the prediction result. Setting \(\lambda\) too small might result in not enough sample points. This can cause problems like over fitting the model. On the other hand setting \(\lambda\) too large yields the historical prediction approach. In Section 6 we test with different values of \(\lambda\) to find the optimal value for this parameter.

### 4. ROUTE RELIABILITY COMPUTATION

In this section we present a classification of existing approaches for computing route pdfs. Towards this end, we identified three basic characteristics of existing algorithms which become the basis of our classification; link representation, time dependency and link correlation. Different algorithms model link travel times as either discrete or continuous distributions, they can consider the network time dependent or static and finally they would either consider correlation between the travel time of different links or look at them independently. Table 1 summarizes the existing work on route pdfs based on the three attributes we presented here. A brief description of each work is presented in Section 7.

To fully evaluate the techniques in Section 3 for computing \(\text{pltts}\), we need to apply each technique to existing algorithms and check the accuracy of the resulting route pdf \(\text{pmfs}\). Also, since there has not been any study to compute \(\text{pltts}\), it has not been possible to compare the performance of these algorithms based on real world datasets. Following, we briefly explain a few classes of the algorithms presented in Table 1 which we later use in our experiments. The selection of the classes is such that it will give us the ability to compare the effectiveness of the attributes used to classify these algorithms. For example by comparing the results of \text{ConStaInd} \text{(Continuous, Static, Independent)} with the results of \text{ConStaCor} we can find out whether considering link correlations increases the accuracy of the final \text{pmf} or not.

**ConStaInd**: Assuming link travel times are normally distributed and independent, the path travel time is also a random variable which is normally distributed with a mean and a variance, given by

\[
\begin{align*}
\text{mean} & = \sum_{i,j} \lambda_{i,j} \cdot c_{i,j}^s \\
\text{variance} & = \sum_{i,j} \lambda_{i,j} \cdot \left( c_{i,j}^s - \text{mean} \right)^2 
\end{align*}
\]

Table 1: Models considered in this work

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[3]</td>
<td>ConStaInd</td>
<td>continuous</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>[4]</td>
<td>DisStaInd</td>
<td>discrete</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>[5]</td>
<td>ConTimInd</td>
<td>continuous</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>[7, 8, 9]</td>
<td>ConStaCor</td>
<td>continuous</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>[10, 11, 12]</td>
<td>DisStaCor</td>
<td>discrete</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>[13]</td>
<td>ConTimCor</td>
<td>continuous</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>[14]</td>
<td>DisTimCor</td>
<td>discrete</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>
\[ \mu_p = \sum_{(i,j) \in p} \mu_{ij} \text{ and } \sigma^2_p = \sum_{(i,j) \in p} \sigma_{ij}^2 \quad (11) \]

where \((i, j)\) represent links on path \(p\).

**DisStaInd:** Given the pmf \(F_{ij}\) of the link travel time for the link \((i, j)\) we can compute the pmf \(J_{sj}\) of the path travel time of path \(p_{si}\) extended by the link \((i, j)\) by using the poisson-multinomial recurrence.

\[ J_{sj}(b) = \sum_{h=0}^{b} J_{si}(b-h)F_{ij}(h), \forall b = 0, \phi, \ldots, \Phi \quad (12) \]

This equation can be computed efficiently by starting with the first link \((s, i)\) of the path \(p_{si}\) and its corresponding pmf \(F_{si}\). The adjacent link \((i, j)\) is considered next. The pmf \(J_{sj}\) is computed according to Equation 12 and the process repeats with the next adjacent link until node \(d\) is reached.

**DisTimInd:** Let \(F^c_{ij}\) be the pmf describing the link travel time of the link \((i, j)\) at time \(t\) and \(J_{sj}\) be the path travel time for a path \(p^c\). We assume that the route starts at \(s\) at \(t = t_c = 0\) then:

\[ J_{sj}(b) = \sum_{h=0}^{b} J_{si}(b-h)F^c_{ij}(h), \forall b = 0, \phi, \ldots, \Phi \quad (13) \]

Similar to Equation 12 we can build an incremental algorithm upon this equation until node \(d\) is reached.

**ConStaCor:** We manage to capture link correlations for the continuous representation by extending Eq. 11 as follows:

\[ \mu_p = \sum_{(i,j) \in p} \mu_{ij}, \quad (14) \]

\[ \sigma^2_p = \sum_{(i,j) \in p} \sigma_{ij}^2 + \sum_{(i,j) \notin p} \rho_{ij} \sigma_{ij} \sigma_{kl} \quad (15) \]

This compact representation of the correlation of link travel times is only possible due to the assumption of normally distributed link travel times.

5. **Evaluating Probabilistic Predictions**

Evaluating the accuracy of probabilistic results is a task that has been either over simplified or brushed under the carpet by many studies, focusing on the technical aspects of managing uncertain data. However, we argue that this part is crucial in order to justify the use of the uncertain aspects in the data, rather than performing the traditional approaches for data cleaning (e.g., using the expected prediction time). The evaluation of probabilistic prediction models for ordinal values (e.g., travel time) is a non-trivial problem. The evaluation of probabilistic prediction models is time dependent. Fields of application are weather forecasting or betting games. The most prominent representative is known as the Brier scoring rule [22], which is described as follows. Let \(f^c\) be a probabilistic prediction model (e.g., a probability mass function) for a variable, whose true value \(v^t\) is one of several classes \(C\) (e.g., tomorrow is sunny (75%) or rainy (25%)), then for one specific \(t\) the Brier scoring rule returns:

\[ S^B_{\text{Brier}}(f^c, v^t) = -\sum_{c \in C} (f^c(c) - I(v^t = c))^2 \]

where \(f^c(c)\) represents the probability that is assigned to class \(c\) and \(I(v^t = c)\) returns 1 if the true value of \(v^t\) is equals \(c\).

Intuitively, this scoring function gives a reward dependent on the probability that the probabilistic prediction model assigns to the true value. Summing up these rewards over several times \(t \in T\) yields a score. The higher such score is, the better the prediction model \(f^c\) predicts the true probability distribution of \(v^t\) at each value of \(t\):

\[ S^B(f, v) = \sum_{t \in T} S^B(f^c, v^t) \]

With this mechanism, it is possible to evaluate two probabilistic prediction models \(f_A\) and \(f_B\) by comparing their corresponding rewards. Scoring rules can also be used in our scenario, however they are originally designed for categorical values (e.g., sunny, cloudy, rainy) rather than ordinal values (e.g., travel time). Thus these scores are not sensitive to distance, meaning that no credit is given for assigning high probabilities to values near but not identical to the one materializing (e.g., the true outcome). For example a probabilistic prediction model \(A\), that predicts 5 minutes (90%) or 6 minutes (10%) and a model \(B\) that predicts 6 minutes (10%) or 10 minutes (90%), both obtain the same score when the true travel time is 6 minutes. However, we argue that this approach does not account for the ordinal nature of travel times since a true travel time of 6 minutes is rather close to the highly probable 5 minutes of model \(A\) and a good scoring rules should thus favor model \(A\) over \(B\). For this reason we propose to use the continuous ranked probability score (CRPS) which is defined as follows [23]:

\[ \text{CRPS}(f^c, v^t) = \int_{-\infty}^{+\infty} \int_{-\infty}^{v^t} f^c(y)dy - I(x \geq v^t)dx \quad (16) \]

CRPS thus expresses some kind of distance between the probabilistic forecast \(f^c\) and truth \(v^t\). Another very useful property of CRPS is that it can be computed in linear time (in the number of possible outcomes) in the case when \(f^c\) is given by a pmf and has a
closed form for the case where \( f^* \) is given by a normal distribution \( \mathcal{N} \) with parameters \( \mu \) and \( \sigma \) [24]:

\[
CRPS(\mathcal{N}(\mu, \sigma), \nu) = \int \left( 2 \varphi \left( \frac{x - \mu}{\sigma} \right) - \frac{1}{\sqrt{\pi}} + \frac{x - \mu}{\sigma} \left( 2 \Phi \left( \frac{x - \mu}{\sigma} \right) - 1 \right) \right) \, dx
\]

where \( \varphi \) and \( \Phi \) denote the probability density function and the cumulative distribution function of a standard Gaussian variable. One advantage of the CRPS is that it reduces to the mean absolute error (MAE) if the forecast is deterministic. In practice, this makes it possible to compare an ensemble forecast with a deterministic forecast of the same variable in a consistent fashion. Since for the experimental evaluation we discretize the time to 1 second (i.e., \( \phi = 1 \sec \)) this means that the CRPS score can be interpreted as the mean absolute error of the predicted travel time in seconds.

6. EXPERIMENTS

6.1 Dataset and Experimental Setting

6.1.1 Dataset

In our research center, at USC’s IMSC, with our partnership with Los Angeles Metropolitan Transportation Authority (LA Metro), we maintain a big transportation data warehouse that fuses and analyzes a very large-scale and high-resolution (both spatial and temporal) traffic sensor data from different transportation authorities in Southern California, including California Department of Transportation (Caltrans), Los Angeles Department of Transportation (LADOT), California Highway Patrol (CHP) and Long Beach Transit (LBT). This data set includes both inventory and real-time data with update rate as high as every 30 seconds for freeway and arterial traffic sensors (9300 loop-detectors) covering more than 3500 miles, incidents such as accidents, traffic hazards and road closures reported (approximately 400 per day) by LAPD and CHP, and ramp meters. We have been continuously collecting and archiving aforementioned datasets for the past three years. We use this real-world dataset to model and evaluate our techniques. To generate time dependent travel times we spatially and temporally aggregate traffic sensor data by assigning interpolation points for each 5 minutes that depicts the travel-times on the segments of Los Angeles road network with 304,162 edges. Based on our observation, all roads are fairly un-congested between 9:00pm and 6:00am, and hence we assume static edge weights between those times. More detailed information regarding the dataset can be found in [25].

6.1.2 Experimental Setting

All experiments were conducted on an Intel(R) Core(TM)2 Duo 3.16GHz PC running on Microsoft Windows 7. Methods were implemented in Java.

Predicting travel times in a rather static road network (with only marginal changes in the link travel times) is not very challenging and usually does not require a probabilistic prediction. For this reason we focused on rush hour times which imply more uncertainty and thus a much harder scenario for prediction. If not mentioned otherwise we used 8:00am, 9:00am, 4:00pm, 5:00pm and 6:00pm during the weekdays. These times were considered as the start time, i.e., the time at which the user starts his route. According to the results from [17] and our own observations, traffic patterns are quite equivalent across weekdays. Consequently, we did not differentiate between predictions for different weekdays. This means for each start time, we consider data of 260 days (5 days per week, 52 week per year).

In our evaluation process, we used the k-fold cross validation (k=5) method to divide our data into test (test days) and training (training days) sets. In each fold, for each start time and each test day we investigated the following:

- We used the data of the training days with one of our techniques to build a model for predicting the travel time of a link (route) at a specific start time. In case the current situation is required for building the model, we can fetch it from the dataset.
- For the specific test day and start time, we find the actual travel time over the link (route) from the dataset.

Both of these outcomes are then used to compute a CRPS score as defined in Section 5, which is then averaged over all folds and trials. For all discrete models we discretize the time into 1 second intervals.

6.2 Probabilistic Link Travel Time Prediction

In our first set of experiments we evaluate the prediction quality of the following three proposed approaches:

- HP: Prediction purely based on historical data as described in Section 3.1.
- LI: Prediction based on linear interpolation between the current situation and the prediction based on historical data as discussed in Section 3.2.1.
- SH: Prediction based on historical data with behavior similar to the current situation as proposed in Section 3.2.2.

We tested all techniques over 50 links which are chosen randomly from all around the road network explained in Section 6.1.1. The links we chose for these experiments are ~1 mile long and all are on highways. The link selection process was such that we had 6-8 links from each major highway in the L.A. metropolitan area. The farther an event is in the future, the harder it gets to predict it. For each start time during the experiments we use different prediction times varying from 0 minutes up to 120 minutes.

With the HP approach, we only consider historic data. This means that if we use 8:00am as the start time, we consider the travel times at 8:00am across the training days and build the model distribution. Subsequently, we retrieve the actual travel time at 8:00am for each test day and compare it with the model. Since the historic approach is based on historical data only and does not depend on current situation (query time), it is consequently independent form the prediction offset (Fig. 4(a)). Fig. 4(b) shows results of the HP approach for different start times in the day with two implications. First, different times of the day are differently hard to predict. In particular the selected standard start times yield much higher CRPS compared to 9:00pm, where there is not too much traffic on the road network and thus not much uncertainty in the prediction. Second, it shows that the continuous and discrete representation performs rather equivalent. The small difference between the discrete and the continuous case can be explained by the fact that a discrete representation copes better with non-normally distributed link travel times. However, it is important to note that the CRPS of the discrete representation is very sensitive to the discretization interval of the time, i.e., a large discretization interval leads to a larger value of the CRPS and hence not directly comparable to the CRPS of the continuous representation.
Since the results for continuous and discrete link travel times are rather equivalent, for the remainder of the experiments on link travel times we only present results for discrete representation of travel times.

With regards to the LI approach, we studied the accuracy of the model for different time horizons $\tau$, ranging from 15 minutes up to 120 minutes. The results in Fig. 5(a) are averaged over all prediction times and clearly show neither of these extremes should be chosen, but rather a good choice of the time horizon is somewhere in between (i.e., around 60 minutes). These results are backed up by the experiments shown in Fig. 5(b). In the LI approach, we eventually use the time horizon to compute $\theta$ as the weight of the current situation versus historic data. In Fig. 5(b) we directly assign different values to $\theta$. If increasing $\theta$ yields better results, it means assigning more weight to the current situation is beneficial and vice versa. Fig. 5(b) shows that for prediction times greater or equal to 60 minutes, increasing $\theta$ results in higher CRPS scores. Hence we can say, the current situation has no positive influence on the prediction after 60 minutes, which is the definition of the time horizon parameter in Section 3.2.1. We use the results in Fig. 5 to conclude that an optimal time horizon in the Linear Interpolation technique is 60 minutes. Let us note that this is an insightful finding in itself. This result suggests, that the traffic “forgets” its condition an hour ago.

In the next set of experiments, we evaluate the accuracy of the SH approach. As explained in Section 3.2.2, SH only considers the historical data similar to the current situation. We also defined similarity by the parameter $\lambda$ in Section 3.2.2. We assign different values to $\lambda$ and compare the results. Fig. 6 shows the results. Like LI, SH predicts the near future more accurately than the far future since it is taking the current situation into account. For smaller values of $\lambda$ (e.g., 0.01 & 0.05) the predictions yield generally higher CRPS. The reason for this behavior is that only very few measurements of the training data meet the similarity requirement and thus the prediction is based on a possibly non-representative sample. Therefore, as the similarity threshold increases, our model gets more realistic and the results get better. On the other hand, if the similarity threshold is too loose, data that is not similar to the test day is considered in generating the model, which results in less accurate models.

Finally, we compared the three approaches to each other, using the best setting for the corresponding parameters ($\tau = 60$ minutes and $\lambda = 0.1$). The results are illustrated in Fig. 7. Clearly, LI and SH are able to predict the near future more accurately than the farther future, whereas HP is independent from the prediction time. As expected, when $\tau = 60$, for prediction times greater than 60 minutes, LI and HP give the same model. In the following experiments we will use the results of Fig. 7 to show the important fact that the accuracy of all algorithms proposed to compute a pmf for a route, directly depends on how accurate the probabilistic link travel times are estimated on such route.

### 6.3 Route Reliability Computation

In the following experiments, we apply the techniques for computing $\text{pltts}$ to the approaches discussed in Section 4. The first goal
is to evaluate the impact of choosing the right pltt computation technique on the accuracy of the route pmf. In addition we will assess the effectiveness of considering the attributes used to classify route computation algorithms in Section 4.

For these experiments, we use 10 different paths, spanning the road network in Section 6.1.1. The selected paths are 45 miles long on average, consisting of approximately 40 links. Each path corresponds to one of the major highways in the L.A. metropolitan area. For each path, we use the same standard start times we used for the link travel time predictions. Again, we build a model with each approach using data from model days and compare the accuracy of the model with the actual travel time during the test days. Finally, when using the pltt computation techniques, we only use the optimal parameter discovered in Section 6.2 for each technique.

As we can see in Fig. 8, depending on how we model the link travel times, the accuracy of the route pdf can vary. Comparing Fig. 8 and Fig. 7 shows that the accuracy of pltt models directly impact the accuracy of route pmfs. In fact, applying a good pltt model to a bad route model can yield more accurate models than applying a bad pltt model to a better route model.

In Fig. 9 we put the results of applying the SH approach (Section 3.2.2) to each route computation algorithm side by side. Several observations can be derived from this experiment:

- The ConStaInd approach is again time-independent and it seems that the consideration of correlation between links does only yield a marginal improvement over the same approach without correlation (ConStaInd). In our experiments, we observe only a 1%-2% improvement through the incorporation of correlation in this setting.

As an additional comparison partner we included an approach that is used in current route planning systems, named Baseline. Baseline uses the situation at the query time and computes the path travel time without prediction based on the currently available deterministic travel times. The outcome will consequently be a non-probabilistic value as well. The CRPS method allows us to directly compare this approach to the probabilistic counterparts, since it converges to the mean error in the case of certain predictions (i.e., the CRPS can be interpreted as the difference of the prediction and the true value in seconds). The results show that under all settings the probabilistic approaches are superior to the Baseline approach which once more justifies the use of probabilistic models in this application.

In order to give the readers a better understanding of how different the travel times computed by each method in Fig. 9 are with the actual travel time, in addition to CRPS, we also compute the Expected Value Difference (EVD) for all methods in Fig. 9. To compute the EVD, we build the route pdf as we did before and instead of computing the CRPS score between the pdf and the actual time, we take the delta of the pdf’s expected value and the actual travel time. These results are illustrated in Fig. 10.

Lastly, we evaluated the runtime of all path computation methods discussed in Section 4. Runtimes are broken into link travel time prediction and path computation in Figure 11. Generally, the continuous approaches perform much better due to the analytical nature which yields a much better runtime complexity. As expected, the more complexity we add to the approaches based on a discrete representation, the more costly in terms of computation they become.

Figure 8: Applying pltt computation techniques to route computation algorithms

Figure 9: CRPS for probabilistic path computation

Figure 10: EVD for probabilistic path computation

Figure 11: Runtimes
7. RELATED WORK

In this section we will review related work on the problem of computing probabilistic link travel times.

7.1 Uncertain Link Travel Times

Uncertainty in databases has attracted a lot of attention from the research community in the last decade. Many approaches for managing [26, 27, 28], querying [29, 30] and analysing [31, 32] these uncertain datasets have been proposed. Especially in the field of spatial [33, 34] and spatio-temporal [35, 36] data, uncertainty was considered as an important issue due to uncertain sensor measurements or probabilistic estimations of unknown values. Although these plethora of works provide many efficient techniques to deal with the uncertainty, only a few works actually consider how to obtain this uncertainty in a way such that the outcome of these techniques is meaningful and evaluate the whole end-to-end pipeline of this process.

The first work [3] dealing with uncertain link travel times in road networks considers a graph with uncertain edge weights with arbitrary distribution. The particular interest of this work is to study the characteristics of the probability distribution describing the length of the shortest path between a source and a destination. The author considers the problem from a statistical point of view and no algorithm (besides a Monte-Carlo driven sampling approach) to compute this distribution is provided.

Time dependency of link travel times was recently studied in large scale traffic networks [37, 17, 38] without uncertainty. It has been shown that considering the predicted value of a link travel time at the arrival at that link rather than its value at start of the route vastly improves the accuracy of fastest route algorithms.

Miller-Hooks and Mahmassani [4, 5] consider addressing the problem of finding the Least Expected Time (LET) paths. They consider stochastic, time-varying networks where link travel times are independent random variables with time dependent (discrete) probability distributions. A different study, Nie and Wu [6] proposed an algorithm for solving the shortest path problem with on-time arrival reliability (SPOTAR). The SPOTAR problem is to find the latest possible departure time and the associated route to attain a given probability of arriving at the destination at a specified arrival time or earlier. The authors model link travel times by independent discrete probability distributions. However, none of these studies propose an algorithm to compute the distributions for link travel times.

In [9] the authors assume normally distributed link travel speeds. The ultimate goal is to find the least expected time path (cf [5]) but between a given source and destination rather than all possible sources. Due to the problem setting, no probabilistic guarantees (reliability) can be assigned to the resulting path.

Link travel times in traffic networks are generally dependent on each other. For example a traffic jam usually extends over several road segments and a high travel time on one segment usually implies a high travel time on the next segment. The consideration of correlation between link travel times of different edges is thus certainly relevant in traffic networks.

In [10] the authors propose an adaptive path finding strategy for the stochastic on-time arrival (SOTA) to provide the user with the route with the highest probability to arrive at the destination before a given time threshold. They use a discrete model to represent link travel times and consider a simple form of correlation between adjacent links in which the travel time of an edge are modelled by two probability mass functions that represent the normal and the congested state. The states of two adjacent links are then correlated by conditional probabilities. Based on the uncertainty model of [10], the authors of [11] proposed an approximation algorithm for the probabilistic path travel time and heuristic approaches to find the best path(s). A similar study [12] presents an algorithm based on dynamic programming in order to return the path from a given source to a destination with the least expected travel time (LET). The link travel times are given by pmfs and correlations are modelled similar to [10].

Another work in this direction is [7], where the authors consider the problem of finding the optimal reliability path (ORP), which is equivalent to the SOTA problem. In their setting the travel time of an edge is normally distributed and travel times of different edges may correlate without changing over time. [8] models link travel times as gamma distributions and provides an approximate solution using a Monte-Carlo based approach to the SPOTAR (cf [6]) problem.

Including both, time-dependency and correlation of link travel times for route optimization problems is generally a very complex undertaking and has only been considered in very recent studies.

In [13], time-dependent road networks with normally distributed link travel times and local correlations between them based on two states (similar to [10]) are considered.

In [14], the authors extend their previous work [6] by introducing limited spatial correlations into the SPOTAR problem - referred to as reliable a priori shortest path problem (RASP). Specifically, the probability mass function of the traversal time on a link is assumed to be conditional on the state of that link. The states of a link have a Markovian property. The probability distribution of link traversal times is also allowed to vary over time, which provides a mechanism to account for the dynamic network behavior.

7.2 Prediction of Link Travel Times

The prediction of link travel times is an essential part for the real time computation of the path travel time in a time-dependent traffic network. Thus, there exist a large body of work that solely concentrates on techniques solving this problem based on single link models [17], traffic incident models [1] or Bayesian network models [39]. Neither of these approaches does however incorporate the inherent uncertainty of the predicted values and does not allow for the use of the algorithms discussed in Section 7.1. Recently Yang et al. proposed to utilize spatio temporal hidden markov models for this purpose. However the construction of the model is costly and the prediction process does not include the current situation in the network which makes the approach hardly applicable in the real-time setting considered in this work.

8. CONCLUSIONS AND FUTURE WORK

In this work we studied the problem of assigning reliability or confidence values to the estimated travel time of a path in road networks. We showed that in order to achieve this goal, a whole pipeline of methods has to be present. We developed three novel methods to obtain probabilistic link travel times for each link in the network, using historical and current traffic observations. We presented several methods for summing up the travel times of each link on a given path under different models. We also introduced a method for evaluating probabilistic link and path travel time estimations. This technique made it for the first time possible to compare the existing approaches and the developed link travel time estimation methods against each other in terms of accuracy.

The results presented in this paper leave room for improvement and further investigation. In the current study, we only consider traffic data from the L.A. metropolitan area. Some of the parameters we compute in our experiments (e.g., the road network forgets the traffic from 1 hour ago) may end up having different values on
road networks in other parts of the globe. It would be interesting to investigate whether such parameters have the same values on other road networks as well and if not what may cause the difference. In addition, a follow up study on the efficiency of calculating/storing/evaluating plts (either via pre-processing or real-time updates) and directly applying them within industrial-grade route-planning algorithms would be of significant practical value in this domain.

Acknowledgement
This research has been funded in part by NSF grant IIS-1111513, Caltrans, the USC Integrated Media Systems Center (IMSC), and unrestricted cash gifts from Google, Northrop Grumman, Microsoft, and Oracle. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of any of the sponsors such as the National Science Foundation.

9. REFERENCES


[21] Pearson, “X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling,” Philosophical Magazine Series 5, vol. 50, no. 302, pp. 157–175, 1900.


[26] Pearson, “X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling,” Philosophical Magazine Series 5, vol. 50, no. 302, pp. 157–175, 1900.


[31] Pearson, “X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling,” Philosophical Magazine Series 5, vol. 50, no. 302, pp. 157–175, 1900.


[36] Pearson, “X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling,” Philosophical Magazine Series 5, vol. 50, no. 302, pp. 157–175, 1900.
