Vehicle Travel Time Prediction Algorithm Based on Historical Data and Shared Location

Peng Chen, Zhao Lu and Junzhong Gu
Department of Computer Science and Technology
East China Normal University,
Shanghai, 200241, China
{pchen}@ica.stc.sh.cn
{zlu,jzgu}@cs.ecnu.edu.cn

Abstract—In recent years, the travel time predictions have become a popular research topic. In this paper, we present a new algorithm of the travel time predictions based on the idea of using the shared traveler’s positions to collect traffic conditions. Several experiments show that our algorithm has a broader applied area than existing algorithms and can provide real-time and the accurate predictions for the travelers. And when there are more travelers and more positions shared among them, the more accurate predictions of our algorithm will be.

Keywords: ATIS, neural network, shared position, route guidance, TP-HDSL

I. Introduction

It is widely believed that providing the travel time information to travelers not only can yield positive effects on the overall time and the cost savings and thus can improve traveling comfort, especially for the travelers who are not familiar with the place they want to go, but also can reduce the traffic congestion in some extent from a macroscopic point of view [3]. But so far, the travel time prediction is still a complex and challenging problem.

Some factors which impact the travel time predictions are irrational, such as traveler’ attention, driving habits, risk assessment and vehicles’ acceleration/ deceleration performance. And the weather, disasters, incidents and accidents also influence the prediction. It is obvious that it is impossible to predict the impact of drivers’ rational or irrational behaviors and all external factors.

Till now, there are mainly two different strands about travel time prediction approaches, model-based approaches and intelligent inductive (data-driven) approaches. The model-based approach uses the traffic information (condition) and some hypotheses on both drivers’ behaviors and the traffic conditions to build a model, predicts the future traffic conditions on top of the model, and calculates the travel time based on the predicted conditions. It needs highly accurate models and accurate inputs to the models [5]. So far, many successful efforts have been reported, including DynaMIT, DynaSMART and BOSS, but still a lot of problems remain unsolved, e.g. the rationality of the hypotheses, the complexity and the generality of the model.

There are so many factors that will impact the travel time prediction, and it is impossible to sample, quantify and tell the effect of each factor. So we pay attention to the intelligent inductive (that is data-driven) approach, including support vector regression [6], feed forward neural network - and recurrent neural network [7], to name a few.

The following sections present an account of our work. In Section 2, we describe a real system in which we implement the proposed algorithm. In Section 3, a detailed description of the travel time prediction algorithm is shown. The evaluation of the proposed algorithm is introduced in Section 4 with some experimental results. Finally, the last section presents the conclusions of the paper.

II. Background

A. LaMOC Systems

LaMOC system is a collaborative platform for location services. It provides personal location information search, recommend services, personalized maps and other services. In LaMOC system, users can provide other users in virtual organizations a wide range of mobile information services conveniently and mutually.

LaMOC has a client-server structure. The client, the application software installed in the mobile device, is in charge of collecting travelers’ location information , managing the request and data transfer between the client and the server and displaying the recommended information. The server is composed of the GIS server [8], the collaboration server, the application server, the historical information database and the route guidance
server. As shown in Fig.1 the route guidance subsystem of LaMOC which implements our algorithm, provides the recommended route service and the predict travel time service.

![Image](image.png)

Figure 1. The top architecture of the route guidance subsystem of LaMOC

The top architecture of the route guidance system and the data flows among its components are shown in Figure 1.

B. THE COLLECTION OF TRUE TRAVEL TIME

Till now, various algorithms proposed use sensors installed on roads (links) to collect all kinds of information about traffic conditions. While in this paper, mobile devices are used as units for collecting users’ positions [9, 10, 11]. In this way, the distance limitation between the sensors is reduced. And the algorithm’s application scope is expanded. The client reports its user’s position in short time periods. we assume that a traveler could not go through two links at normal speed during such a period of time. the privacy problem of positions shared among users is beyond the scope of this paper.

III. TRAVEL TIME PREDICTION ALGORITHM BASED ON HISTORICAL DATA AND SHARED LOCATION

There are three merits of neural networks which play an important role in our choice. First, the neural network has an ideal adaptability to changes. Second, it is capable of dealing with complex non-linear spatio-temporal relationships between the observable traffic quantities. Third, over the course of many examples being presented to the neural network, the complex dynamic properties of traffic flow are learned directly from data.

From the previous analysis, we can come to the conclusion that the neural network is capable of link travel time prediction. Then we use it in the TP-HDSL algorithm.

A. THE PROCEDURE OF TP-HDSL

The TP-HDSL algorithm can provide its users with route guidance advice and travel time information. It can be described as follows:

- According to the user’s current position, his destination, the road topology information, historical travel times and recent travel times of links, several candidate routes are calculated. The link travel times of all the links in that candidate route are also calculated. Furthermore, the total travel time of each candidate route will be calculated.
- While the user is moving along the recommend route, the predicted travel time will be adjusted according to accidents, the route conditions, the deviation between user’s position change and prediction, etc. When traffic conditions change considerably, we will inform the user and offer a better route. This procedure will continue till the user reaches his destination.
- Go to next route recommendation.

B. THE TRAVEL TIME PREDICTION FOR THE LINKS

In this paper, the path between two adjoining intersections is called a link. It is the smallest unit of the travel time prediction. The link travel time is defined as.

\[
x_j(t) = G(F(j), X_t, t, HX, EX_j(t))
\]

Where \( x_j(t) \) denotes the time prediction of link \( j \) at time instant \( t \). \( F(j) \) is the basic information of link \( j \). \( HX \) denotes the actual travel times before the time instant \( t \). \( EX_j(t) \) is the total effect of the weather, the traveler’s health condition and other uncertain factors. \( X_t \) represents 10 groups of historical travel times used in the prediction at time instant \( t \). Historical travel times are the long-term statistic values of actual travel times for the link. They reflect the average congestion level of the link at instant \( t \). Consider both usage and maintenance, 10 groups of historical travel times are used. 10 groups of data are enough to give accurate prediction and achieve a reasonable database update rate. A further increase of the group size will not improve the accuracy noticeably. \( F(j) \) and \( HX \) have been learned by training the neural network.

The structure of the neural network model used in link travel time prediction is schematically outlined in Figure 2.
is a function of all the weights, biases and
\( t \) is the number of
with respect to \( t \) is calculated.
\[
\frac{\partial t}{\partial t} = t(t)\Phi(t)
\]

The system will
update the weights and historical travel
time according instan
ton.

1) THE INPUT OF THE NEURAL NETWORK

In Fig 2, the time instant \( t \), 10 groups of historical
travel times and the external factors at instant \( t \) are
taken as the network’s input.

In order to predict link travel time at any instant,
half-hour historical traffic conditions before and after
that instant will be referred to. When a prediction is
made at time instant \( T' \), historical travel times at
instants \( T + \Delta t \) \( \ldots \) \( T + 5\Delta t \) \( \ldots \) \( T + 10\Delta t \) are
used in the prediction, where \( T + 5\Delta t < T' < T + 6\Delta t \).

2) MATHEMATICAL STRUCTURE OF THE NN MODEL

A logistic sigmoid function is used as the hidden
layer transfer function, and identity function is used as
the output layer transfer function.

The input layer is partially connected to the hidden
layer. The relation between the input layer and the
hidden layer can be defined as follows
\[
\begin{align*}
\Phi(t) = & \Phi_w(v_i(t)) \\
\end{align*}
\]

\[
\begin{align*}
\frac{\partial t}{\partial t} &= \Phi(t)
\end{align*}
\]

Where \( y_i(t) \) denotes the output of neuron \( i \) on
the first hidden layer at instant \( t \). \( w_{i0}^{1} \) denotes the bias of
neuron \( i \) on the first hidden layer. \( M \) is the number
of neurons in the input layer. \( u_m(t) \) denotes the input of
neuron \( m \) on the input layer at instant \( t \). \( w_{im}^{1}(t) \) depicts
the weight from neuron \( m \) on the input layer to neuron \( i \)
on the first hidden layer at instant \( t \). \( N \) is the number of
neurons in the first hidden layer. In the hidden layers,
the neurons on \( j \)th hidden layer are fully connected to
the neurons on \( (j+1) \)th hidden layer.

\[
\begin{align*}
v_i^{(j+1)}(t) &= (w_{i0}^{1} + \sum_{m=1}^{M} w_{im}^{1}(t)\Phi(u_m(t)))(i \in [1, N])
\end{align*}
\]

\[
\begin{align*}
\frac{\partial t}{\partial t} &= \Phi(t)
\end{align*}
\]

The relation between the neurons on adjoining
hidden layers is defined in (4) and (5). Where \( y_i^{(j+1)}(t) \)
denotes the output of neuron \( i \) on the \((j+1)\)th hidden
layer. \( w_{i0}^{j+1} \) is the bias of neuron \( i \) on the \((j+1)\)th hidden
layer. \( N \) is the number of neurons on the \( j \)th hidden
layer. \( y_n^{(j)}(t) \) is the output of neuron \( n \) on the \( j \)th hidden
layer at instant \( t \). \( w_{in}^{j}(t) \) denotes the weight from
neuron \( n \) on the \( j \)th hidden layer to neuron \( i \) on the
\((j+1)\)th hidden layer at instant \( t \).

where \( L \) denotes the number of hidden layers. The
\( L \)th hidden layer is fully connected to the output layer.
The relationship between them is defined as
\[
\begin{align*}
y_i(t) &= (w_{i0}^{1} + \sum_{n=1}^{N} w_{in}^{j}(t)\Phi(u_n(t)))(i \in [1, N]) + 1 \\
\end{align*}
\]

\[
\begin{align*}
\frac{\partial t}{\partial t} &= \Phi(t)
\end{align*}
\]

The TP-HDSL algorithm runs in batch training
mode. When 20-days data are used in training, the
neural network can correctly reflect the real traffic
conditions. A longer training time will not significantly
improve the accuracy.

The training algorithm aims at reaching a minimum
of \( E(t) \) defined in (8).

\[
\begin{align*}
E(t) &= \frac{1}{2} \sum_{j \in E} e_j^2(t)
\end{align*}
\]

\[
\begin{align*}
\frac{\partial t}{\partial t} &= \Phi(t)
\end{align*}
\]

\( E(t) \) is a function of all the weights, biases and
inputs. During the training, all the weights will be
adjusted and the output will be moved towards the
desired output.

The training procedure is described as follows. First,
the gradient of \( E(t) \) with respect to \( w_{in}^{j} \) is calculated.

\[
\frac{\partial E(t)}{\partial w_{in}^{j}} = \frac{\partial E(t)}{\partial e_j(t)} \frac{\partial e_j(t)}{\partial y_j(t)} \frac{\partial y_j(t)}{\partial w_{in}^{j}}
\]

Then, let \( \delta_j(t) = -\frac{\partial E(t)}{\partial y_j(t)} \frac{\partial y_j(t)}{\partial w_{in}^{j}} \)

\[
\begin{align*}
\frac{\partial t}{\partial t} &= \Phi(t)
\end{align*}
\]
Finally, \( \Delta w_{ji}^l(t) = -\eta \delta_j^l(t) y_j^{l-1}(t) \)

Where \( \Delta w_{ji}^l(t) \) is the adjusted value of \( w_{ji}^l \). \( \eta \) is the learning step.

Because different transfer functions are used, the local gradients for the output layer neurons (\( \delta_j^l(t) \)) and those for hidden layer neurons (\( \delta_j^l(t) \)) are calculated in two slightly different ways.

The local gradient for output layer neuron (\( \delta_j^l(t) \)) is expressed as
\[
\delta_j^l(t) = e_j(t) = a_k(t) - y_k(t)
\]

The local gradients for hidden layer neurons is defined as
\[
\delta_j^l(t) = \Phi(\delta_j^l(t)) = \sum_k \delta_{kj}^{l+1}(t) w_{kj}^{l+1}(t) = e_j(t)\big[1-\delta_j^l(t)\big] \sum_k \delta_{kj}^{l+1}(t) w_{kj}^{l+1}(t)
\]

The weight will be changed according to the following equation.
\[
w_{jk}^l(t+1) = aw_{jk}^l(t) + \eta \delta_j^l(t) y_k^{l-1}(t) (a \in (0,1))
\]

And \( a \) is a useful parameter. After several attempts, a reasonable \( a \) is found, which can increase the speed of adjustment and make the adjustments more stable.

C. TRAVEL TIME PREDICTION FOR ROUTES

In this paper, the path between the start point and the destination is called a route. It is consist of several links. \( X_1, X_2, \ldots, X_{m-1} \) are used to denote the travel time predictions of the links in the recommended route.

The recent link travel time means some of our users actually went through the link at some moment immediately before the prediction instant. This value can reflect the traffic condition more accurately. Depending on whether or not we can utilize the recent link travel time the calculation formulas will be different.

(1) Without recent link travel time, the travel time of the route is defined as
\[
X = \sum_{i \in M} X_i
\]

Here, \( M \) denotes the set of links in the route and \( X_i \) is the predicted link travel time.

(2) With recent link travel time, the travel time of the route is defined as
\[
X = \sum_{i \in P} X_i + \sum_{j \in Q} (aR_j + bX_j)
\]

Where \( a > b \) and \( a + b = 1 \).

Here, \( P \) denotes the set of links without recent link travel times, and \( Q \) denotes the set of links with recent link travel times. Where \( P \cup Q = M \). \( R_j \) is the recent travel time of link \( j \).

(3) Finally, the route travel time prediction is defined as
\[
T_{sum} = X + k
\]

Here, \( k \) is a random number. Users are more likely to accept the calculation if the predicted travel time is longer than the actual travel time. Otherwise, they may doubt the accuracy of the prediction.

IV. RESULTS ANALYSIS

Prototype Implementation: The neural network and TP-HDSL algorithm is implemented with the Neural Network Toolbox of Matlab6.

Setup: The experiments were performed on a 1.66GHz/1GB Intel(R) running Windows XP and Matlab6. The client is Dopod P660 cell phone with GPS module.

Simulating environment: The route is consisted of 10 links. The probability of having the recent travel times is one-tenth.

Generation of data: the data set contains 40-day traffic conditions is generated and later been used in both training and evaluating. First, 20-day data is used as the training set, and then another 20-day data as testing set. The RME of the data set we use is about 10%.

Performance index: RME (Relative Mean Error) is applied as the performance index
\[
\text{RME} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{d_i - \overline{d}}{d_i} \right|
\]

Here, \( \overline{d} \) denotes the average value of the data set.

A. The result of link travel time prediction

The travel time prediction of a link is depicted in Fig.3. The vertical axis is the travel time of the link; the horizontal axis is the departure time at the beginning of the link.

Figure 3 A comparison with prediction and actual travel time of the link with a low probability of congestion.
Analysis: It can be seen in Fig. 3 that, through training and adjusting, predictions can vary with the real traffic conditions and accurately predict them. The accuracy is about 95%.

B. The result of route travel time prediction

The travel time prediction of a route which is consist of ten links is depicted in Fig. 4. The vertical axis is the travel time of the link; the horizontal axis is the departure time at the beginning of the route.

![Figure 4: A comparison with prediction and actual route travel time.](image)

TABLE I. A COMPARISON WITH TP-HDSL AND OTHER ALGORITHMS

<table>
<thead>
<tr>
<th>Interval between sensors</th>
<th>RME</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVR</td>
<td>1km 4.24%</td>
</tr>
<tr>
<td>SSNN</td>
<td>300m 2.37%</td>
</tr>
<tr>
<td>Current-time Predictor</td>
<td>1km 10.41%</td>
</tr>
<tr>
<td>Historical-mean Predictor</td>
<td>1km 14.12%</td>
</tr>
<tr>
<td>TP-HDSL</td>
<td>No limitation [2.43%, 4.48%]</td>
</tr>
</tbody>
</table>

The prediction accuracy of links is about 95%.

Analysis: It can be seen in Fig. 4 that the prediction accuracy of routes are higher than that of links (97.5%-95.5%). Our experiments are conducted under the assumption that there is no obvious relationship between the traffic change of one links and that of another. In the majority of cases, this assumption holds. While specific conditions need specific analysis. The result showed in Fig. 4 is under the assumption that the possibility of getting recent travel time is one-tenth. Actually, the possibility may be higher, and the prediction may be more accurate.

C. Comparison with other methods

To evaluate the applicability of travel-time prediction with TP-HDSL, some common baseline travel-time prediction algorithms (such as SVR, SSNN, Current-time Predictor and Historical-mean Predictor) are exploited for performance comparison. All the algorithms were compared in two aspects (RME and the interval between sensors). Among these algorithms, Current-time Predictor and Historical-mean Predictor are model-based algorithms, while SVR, SSNN and TP-HDSL are intelligent inductive (data-driven) algorithms.

1) SVR

The TP-HDSL algorithm is compared with The SVR prediction algorithm [4].

2) SSNN

The TP-HDSL algorithm is compared with SSNN (State Space Neutral Networks) [8].

(3) Current-Time Predictor

\[ T(t, \Delta) = \sum_{i=0}^{L-1} \frac{x_i + 1 - x_i}{\Delta} \]

where \( \Delta \) denotes the delta delay. \( L \) is the number of links. \( (x_i + 1 - x_i) \) denotes the length of a link.

\( v(x_i, t - \Delta) \) denotes the speed at the start of the link.

(4) Historical-Mean Predictor

\[ \overline{T} = \frac{1}{w} \sum_{i=1}^{w} T(i, t) \]

where \( w \) is the number of weeks trained and \( T(i, t) \) is the past travel time at time \( t \) of historical week \( i \).

It can be seen in Table 1 that we can lower the requirement of road information, and still provide the user with accurate predictions.

V. Conclusion

Our algorithm is a general solution to travel time prediction. Its design is based on the route of interest, and closely related to the real-road condition. Using mobile devices as probes for collecting users’ positions makes TP-HDSL algorithm suitable for links without sensors. When there are more travelers and more positions shared among them, the more accurate our prediction will be.

The privacy problem of positions shared among users can be alleviated to some extent, because the position does not need to be combined with the user’s identification.

ACKNOWLEDGMENTS

This paper is partially supported by NNSFC (No. 90718004 and No. 60703004) and STCSM project (No. 075107006). The authors wish to thank Professor Erich Neuhold and other anonymous reviewers for their valuable comments. We are also grateful to all project members of LaMOC.

REFERENCES


