Improving the accuracy of travel time estimates using archived ITS data

Sirisha M. Kothuri*, Kristin Tufte, Soyoung Ahn and Robert L. Bertini
Department of Civil and Environmental Engineering
Portland State University P.O. Box 751, Portland, Oregon 97207-0751
Phone: 503-725-4285 Fax: 503-725-5950 Email: skothuri@cecs.pdx.edu

Abstract

Advanced Traveler Information Systems (ATIS) are integral components of Intelligent Transportation Systems (ITS). These systems are aimed at providing transportation system users with pre-trip or en-route travel information, thereby allowing travelers to make informed decisions about their travel choices. Most urban travel occurs on freeways; hence, the accurate prediction of travel time over freeway networks has become necessary. In order to develop benchmarks for improved travel time estimates in the Portland, Oregon (USA) metropolitan region, the objective of this paper is to evaluate several travel time algorithms using archived loop detector data and ground truth (probe vehicle) data.

Keywords traveler information, travel time, archived data

Introduction

Improving the accuracy of travel time estimates is essential, especially when used for providing real time estimates to the public via variable message signs (VMS), as shown in Figure 1. Travel time estimates can be generated using different technologies such as inductive loop detectors, video detection, license plate matching, cellular phone matching and various other technologies [1]. The most popular method is to generate travel time estimates using data from inductive loop detectors.

Figure 1: Sample VMS Display

Travel times for different freeway links (corridors) were estimated using two algorithms, the Midpoint and Coifman algorithms, and compared to two ground truth data sets. The algorithms were used to estimate travel time using archived loop detector data from PORTAL – the Portland Oregon Regional Transportation Archive Listing, an Archived Data User...
Service (ADUS) database that stores loop detector data from the Portland metropolitan area at a 20-second granularity [2]. These data are received from the Oregon Department of Transportation’s (ODOT) Traffic Management Operations Center (TMOC). To assess the performance of the algorithms, we compared the accuracy of the travel time estimates generated by the Coifman and Midpoint algorithms to ground truth data. The ground truth data consisted of probe vehicle data and bus dispatch data. The probe vehicle data (90 runs, 15 hours) were collected at multiple links in the spring of 2005 [3]. The bus dispatch data from express buses traversing a key freeway corridor over a 3-week period were obtained from TriMet, the transit agency in Portland.

![Freeway Map Showing Analyzed Links](image)

**Figure 2: Freeway Map Showing Analyzed Links**

**METHODOLOGY**

Travel time estimates from the archived loop detector data were generated using a standard midpoint algorithm and one developed by Coifman. The algorithm proposed by Coifman uses traffic flow theory to estimate travel times for a link [4]. Coifman proposed that the velocity of a vehicle can be represented by:

\[
v(x,t) = f(x + ut)
\]

where \(x\) is the distance, \(t\) is time and \(u\) can be either \(u_f\) the free flow signal velocity or \(u_c\) the congested signal velocity. The vehicle trajectories in a time space diagram can be represented by the differential equation:

\[
\frac{dx}{dt} = v(x,t)
\]
The vehicle’s travel time across a link is the time taken by the corresponding trajectory to travel across the link and is shown in Figure 3. Travel time for a link can be estimated using vehicle velocity \(v_j\), headway \(h_j\) and congested signal speed using the relationship:

\[
\tau_j = \frac{h_j}{v_j} \left(1 + \frac{v_j}{u_c}\right)
\]

\(x_j = v_j \tau_j\) \hspace{1cm} (4)

By measuring headway \(h_j\) and velocity \(v_j\) from the loop data, and using equations (3) and (4), \(t_j\) and \(x_j\) can be calculated.

The successive \(x_j\)s are added to obtain the link distance. However, the total distance obtained can exceed the link distance. Therefore, to accurately estimate the link distance, a weight \(p\) is calculated as shown in equation 5:

\[
p = \frac{x_{k+N_k} + \sum_{j=k}^{k+N_k} x_j}{x_{k+N_k}+1} - d
\]

\(5\)

Finally travel time is estimated as:

\[
T_k = p \cdot \tau_{k+N_k} + \sum_{j=k}^{k+N_k} \tau_j
\]

\(6\)

Figure 4 displays the mechanism of ODOT’s midpoint algorithm currently used for travel time estimation and for dynamic message sign display. Each detector has an influence area as shown in Figure 4. It is assumed that the detector station is at the midpoint of each influence area. Travel time for each segment is calculated as the ratio of the segment length to
measured speed. Travel time is estimated for each segment at each 20-sec interval and aggregated over the entire link.

![Figure 4: Midpoint Algorithm Travel Time Estimation](image)

Six scenarios were tested using the Coifman and midpoint algorithms. The Coifman algorithm was evaluated using either upstream or downstream speeds or a combination of both. The midpoint algorithm was tested in its standard form as well as taking an average of the upstream and downstream detector speeds to generate travel time estimates. These six methods were run on the archived loop detector data in PORTAL and travel time estimates were produced. These estimates were then compared to the ground truth data. These six different are explained below with the help of Figure 5. A hypothetical link of distance $d$ is represented in Figure 5 with upstream detector $a$ and speed $V_a$ and downstream detector $b$ and speed $V_b$. A hypothetical vehicle is assumed to travel along the link and is leaves upstream detector $A$ at time $t=0$ and is at position $P$, somewhere along the link at time $t=t_P$.

The six different methods that were tested are:

- **Coifman algorithm using speeds from upstream detectors only**: $V_{P,t=t_P} = V_{a,t=t_P}$
- **Coifman algorithm using speeds from downstream detectors only**: $V_{P,t=t_P} = V_{b,t=t_P}$
- **Coifman algorithm using speeds from both upstream and downstream detectors using the midpoint influence areas**:
  - $V_{P,t=t_P} = V_{a,t=t_P}$ while $V_{a,t=t_P}t_P < d/2$
  - $V_{P,t=t_P} = V_{b,t=t_P}$ while $V_{a,t=t_P}t_P > d/2$
- **Coifman algorithm using speeds from upstream and downstream detectors weighted in the ratio of distance of the hypothetical vehicle from each detector**:
  $$V_{P,t=t_P} = \frac{\left[V_{a,t=t_P} \times (d - V_{a,t=t_P}t_P)\right] + \left[V_{b,t=t_P} \times (V_{a,t=t_P}t_P)\right]}{d}$$
- **Midpoint algorithm**: $V_{P,t=t_P} = V_{a,t=t_P}$ or $V_{P,t=t_P} = V_{b,t=t_P}$ depending on the influence area.
- **Midpoint algorithm using speed at time (t = 0) that is an average of the upstream and downstream detector readings**: $V_{P,t=t_P} = \frac{V_{a,t=t_P} + V_{b,t=t_P}}{2}$
ANALYSIS

The Coifman and the midpoint algorithms were estimated on archived ITS data from two different time periods. Travel time estimates were derived using the algorithms on archived loop detector data for specific days in 2005 and these estimates were compared to the probe vehicle travel times. In addition, travel times estimated on archived loop data from November 2002 were also compared to the TriMet bus probe travel times.

Probe Vehicle Data Analysis

Mean travel times (and standard deviations) for the probe vehicle and the mean estimates obtained from the Coifman and Midpoint algorithms are shown in Figure 6. From the figure it is apparent that both the Coifman and Midpoint algorithms generate reasonably accurate travel time estimates for most links. The root mean square error, which represents the error between the algorithm generated travel time estimates and the probe vehicle travel times, in almost all cases was lowest for the Coifman u/s travel time estimates. The root mean square was computed for each link by calculating the difference in between the probe vehicle travel times and the estimated travel times for each run. These individual errors were squared and summed and divided by the number of individual errors. Finally, the square root was taken and the root mean square error for each link was produced. The travel time estimates also indicate that the Coifman algorithm (u/s or d/s) produces more accurate travel time estimates than the midpoint algorithm as indicated by the lower root mean square error. The type of Coifman algorithm (u/s or d/s) that generates more accurate travel time estimates depends on the location and formation of queue with respect to the detector. If the queue forms closer to either detector, then the travel time estimates using readings from that detector will tend to be more accurate. However if the queue forms at the midpoint of the link, then using readings exclusively from the upstream or downstream detectors may not produce accurate travel time estimates. In such cases, it may be beneficial to implement the Coifman algorithm using a combination of speed readings from the upstream as well as downstream detectors.
The travel time estimates derived from the algorithms were also tested during a variety of traffic conditions namely free flow and congestion in the aftermath of an incident. All the algorithms produced accurate estimates during free flow conditions. Comparing the travel time estimates produced after the occurrence of an incident revealed that all the algorithms underestimated travel time. Therefore, any algorithm that is used for travel time estimation needs to incorporate the ability to handle incidents and predict travel times accurately especially after the incident has occurred. The accurate prediction of travel time can assist the travelers in choosing alternate routes and minimizing the total system delay.

Probe vehicle, Coifman, and midpoint trajectories were also constructed for Link 6, which represents a section of I-5 just south of downtown. This part of the freeway is characterized by a nonstandard geometric feature known as the Terwilliger curves. In addition to the horizontal and vertical curves, the detector spacing is also large (~ 3 miles). It has been observed that travelers slow down as they approach the curves and accelerate after passing the curves. Since there is no detector present at the exact location where deceleration and acceleration occur, the midpoint algorithm is not able to capture these effects, therefore resulting in a large error. The Coifman algorithm (u/s) produced the most accurate travel time estimate.

**Figure 6: Probe Vehicle Travel Time Estimate and Variance Comparisons.**

**Tri-Met Bus Data Analysis**

Travel time estimates were generated using the six different methods described in the previous sections using the archived loop detector data from November 2002. These were
compared to the Trimet bus travel times for route 96 which covered a section of I-5 N and are shown in Figure 7.

![Figure 7: Bus Travel Time Estimate and Variance Comparison](image)

The root mean square errors for the travel time estimates indicate that the Coifman travel time estimates were more accurate than the midpoint estimates. As indicated in the previous sections, the type of Coifman algorithm that produces the most accurate estimates depends on the location and formation of queue with respect to the detector station.

**CONCLUSIONS**

The goal of this study was to analyze the performance of the standard ODOT midpoint algorithm along with different modifications of Coifman’s algorithm. Travel time estimates produced using these algorithms were compared to ground truth travel times. The results indicate that all the algorithms generate accurate travel times during free flow conditions. While the Coifman algorithm implemented using the upstream detector speeds produced the best estimate during the period after an incident occurred, errors still exist in all estimates produced in the aftermath of an incident and all travel time estimates were underestimated. Travel times were also analyzed for accuracy on links with large detector spacing. Again, the Coifman algorithm with the upstream detector speeds was found to produce the best travel time estimate. Based on the above analysis, the Coifman algorithm was found to be more accurate than the midpoint algorithm.
ACKNOWLEDGEMENT

The authors thank the Oregon Department of Transportation and TriMet for providing the data, the National Science Foundation for funding the research, and the volunteer probe vehicle drivers. Jaspreet Anand pioneered the testing of Coifman’s algorithm using 20 second loop data. The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The National Science Foundation assumes no liability for the contents or use thereof. The contents do not necessarily reflect the official views or policies of the National Science Foundation. This report does not constitute a standard, specification, or regulation.

REFERENCES