Machine Learning Approach to Decomposing Arterial Travel Time Using a Hidden Markov Model with Genetic Algorithm

Shu Yang, Ph.D.[1]; Ming Chen[2]; Yao-Jan Wu, Ph.D., P.E., M.ASCE[3]; and Chengchuan An[4]

Abstract: Bluetooth-based traffic detection is an emerging travel time collection technique; however, its use on arterials has been limited due to several challenges. In particular, data missing not at random (MNAR) is a common data set problem caused by system network failure or sensor malfunctioning. Solving the MNAR problem requires travel-time decomposition (TTD) using complete travel times spanning successive links. Previous work has focused on TTD methodologies that use probe vehicle data. However, these approaches may be unsuitable for Bluetooth-based data. Therefore, this study proposes a machine learning–based approach to decomposing Bluetooth-based travel time. A modified hidden Markov model was developed to model travel-time distributions and traffic-state transitions. A genetic algorithm (GA) was applied to solve a numerical optimal decomposition based on maximum likelihood. Two real-world travel-time data sets were used for validation of the approach. The proposed hidden Markov chain with GA (HMMGA) approach and Gaussian mixture model with GA (GMMGA) were compared with a benchmark approach using distance-based allocation. The results showed that the HMMGA significantly outperformed both the GMMGA and benchmark approaches. Using the HMMGA, the average mean absolute percentage error was up to 72% lower compared to the benchmark approach. **DOI:** **10.1061/(ASCE)CP.1943-5487.0000748.** *© 2018 American Society of Civil Engineers.*

Author keywords: Bluetooth-based travel time; Missing not at random; Travel-time decomposition; Gaussian mixture model; Hidden Markov model; Genetic algorithm.

Introduction

Travel-time estimation plays a vital role in traffic operations and management. Furthermore, the public relies on accurate and reliable travel times to make better trip decisions. Historically, travel times have been collected or estimated based on data from various dedicated traffic sensors (e.g., loop detectors, probe vehicles, and video cameras). Recently, travel-time collection using Bluetooth-based traffic detection systems has become increasingly common. Because of their relatively high accuracy, low cost, and ease of use, Bluetooth systems have been recognized as a valuable traffic data collection technique (Quayle et al. 2010; Bhaskar and Chung 2013; Mohammad and Ashish 2013). Bluetooth sensors can record time stamps and the media access control (MAC) addresses of Bluetooth-enabled devices (e.g., vehicles and smartphones) within a certain detection range. Travel time is then determined by matching the anonymous MAC addresses recorded by sensors at different locations. Bluetooth-based travel times have been verified as high quality data and can serve as ground truth sources (Haghani et al. 2010).

The potential utility of Bluetooth systems to monitor signalized arterial traffic has been shown in several recent studies (e.g., Quayle et al. 2010; Mohammad and Ashish 2013; Day et al. 2010). However, several challenges must be addressed before widespread use of Bluetooth systems in real-world applications can occur. One of the biggest challenges that remains is mitigating missing data. Missing data can have a variety of causes, including low sampling rate, random detection mechanism (Bhaskar and Chung 2013), network communication errors, and sensor malfunction. Some issues such as sensor malfunction can result in a complete loss of travel times over successive links. This problem is referred to as data missing not at random (MNAR) (Schafer 1997). For MNAR, Bluetooth MAC addresses may not be detected over a sequence of successive intersections. Travel times may only be available between the first and last intersections along a corridor, because MAC address detections in the middle links are missing. A common solution to the MNAR problem is to estimate travel times on middle links based on the observed complete travel time. This approach is known as travel-time allocation or travel-time decomposition (TTD) (Hellinga et al. 2008; Hofleitner and Bayen 2011).

Practitioners and researchers can use TTD to estimate a travel time of interest, such as an intersection-to-intersection travel time. Neglecting proper TTD techniques or using an overly simplistic decomposition approach (e.g., proportional allocation by link lengths) may lead to biased results. The usefulness of TTD is not limited to MNAR applications and may potentially have many uses with various traffic data feeds (e.g., probe data and video data). However, most previous studies have only discussed using TTD for processing

[1] Postdoctoral Scholar, Center for Urban Transportation Research, Univ. of South Florida, 4202 E. Fowler Ave., CUT 100, Tampa, FL 33620. E-mail: shuyang@usf.edu

[2] Graduate Research Assistant, Dept. of Civil Engineering and Engineering Mechanics, Univ. of Arizona, 1209 E. 2nd St., Tucson, AZ 85721. E-mail: mingchen7@email.arizona.edu

[3] Assistant Professor, Dept. of Civil Engineering and Engineering Mechanics, Univ. of Arizona, 1209 E. 2nd St., Tucson, AZ 85721 (corresponding author). ORCID: https://orcid.org/0000-0002-0456-7915. E-mail: yaojan@email.arizona.edu

[4] Ph.D. Candidate, Intelligent Transportation Research Center, Southeast Univ., 2 Sipailou Nanjing, Jiangsu 210096, China. E-mail: anchengchuan@gmail.com

Note. This manuscript was submitted on October 10, 2016; approved on September 19, 2017; published online on January 31, 2018. Discussion period open until June 30, 2018; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Computing in Civil Engineering***, © ASCE, ISSN 0887-3801.

probe vehicle data (Hellinga et al. 2008; Hofleitner and Bayen 2011; Zheng and Zuylen 2013; Jenelius and Koutsopoulos 2015). Therefore, proposing TTD solutions for processing Bluetooth-based data becomes necessary. Intuitively, proportioning travel times by link lengths could be a mathematically simple TTD solution if traffic statuses on successive links remain unknown. However, this simple solution could result in less accurate proportions without incorporating link traffic status. Properly describing the relationship of traffic status on successive links serves as the first step to mathematically proportion link travel times. Gaussian mixture models (GMM) and hidden Markov models (HMM) have been widely used to describe the relationship of traffic status on successive links. Genetic algorithms (GA) with constraints are able to solve the TTD problem with the relationship of traffic status presented. Therefore, a combination of either GMM (or HMM) and GA with constraints could potentially improve the TTD accuracy.

The rest of this paper is organized as follows. First, previous studies regarding the methodology of TTD using probe vehicle data and the differences between probe vehicle and Bluetooth-based data are discussed. Next, the proposed methodology for using TTD with Bluetooth-based data is introduced. The proposed methodology consists of a distance-based model, a GMM, a HMM, and GAs. The distance-based model serves as the benchmark approach for comparison, the GMM-based approach is used to model traffic states, and the HMM-based approach is used to model spatial correlation. Next, data description and experiments along Speedway Boulevard in Tucson, Arizona, are introduced. Finally, the results from the experiments are discussed in the Conclusions section.

Previous Studies and Problem Statement

A study by Hofleitner and Bayen (2011) noted that TTD has attracted increasing attention as the importance of estimating link-based and segment-based travel time has been realized. A few studies have investigated TTD using probe vehicle data; in most of these studies (e.g., Neumann 2014; Zheng and Zuylen 2013), probe vehicle data has been evaluated as GPS data points. Fig. 1(a) illustrates TTD in the context of GPS point data. Although previous studies have proposed analytical, probability-based, and artificial neural network (ANN) solutions to determine TTD, the foundations of these studies have relied on vehicle trajectories constructed from sampled GPS data points. For example, Neumann (2014) measured the accuracy of a distance-based TTD solution by utilizing GPS point data collected from multiple probe vehicles on identical road segments. He compared the real-world and estimated vehicle trajectories, and then provided a detailed accuracy measurement for distance-based TTD.

Hellinga et al. (2008) divided link travel time into three components: free-flow travel time, stopped time caused by signal control, and delay probabilities due to congestion. Hellinga et al. then proposed a heuristic method to address the TTD problem, with each component of travel time evaluated individually and their summation determining the total link travel time. In comparison, probabilistic-based approaches are more popular because they utilize joint distributions to capture the relationship between observed data (e.g., travel time) and dependent variables (e.g., traffic states) (Hofleitner and Bayen 2011). For example, Hofleitner and Bayen (2011) developed a decomposition algorithm based on a traffic flow model by modeling link travel times as univariate Gaussian mixture distributions. Each hidden traffic state was represented by a Gaussian distribution. Using Gaussian mixture distributions to model travel-time distribution is reasonable because previous studies (e.g., Yang et al. 2017; Yang and Wu 2016; Jenelius and Koutsopoulos 2015; Feng et al. 2014; Guo et al. 2011) have shown that they can effectively determine travel-time distributions when traffic is heterogeneous. Hofleitner and Bayen's algorithm (2011) then used numerical optimization methods to achieve optimized TTD. Recently, Zheng and Zuylen (2013) used an artificial neural network for TTD. The input to the first neuron layer consisted of the probe vehicle position, link number, time stamp, and speed information. However, the network's applicability was limited to the use of simulation data.

The studies mentioned above show that the issue of TTD with probe vehicle data has been the focus of recent attention. However, TTD approaches using probe vehicle data may be unsuitable for use with Bluetooth-based data. Primary reasons include the data collection mechanism and differences between information collected from probe vehicle and Bluetooth-based sensors. Fig. 1(b) shows a segment with three intersections. Each intersection is equipped with a Bluetooth MAC reader. A moving Bluetooth-enabled device is detected at the first and third intersections but is not detected by the reader in the middle. Unlike GPS devices, which actively report

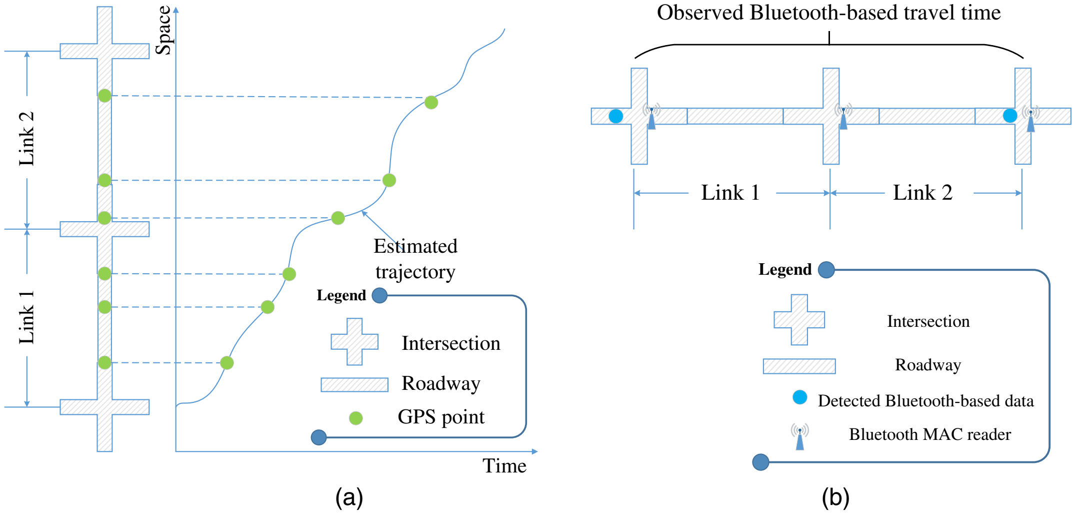

Fig. 1. Travel-time decomposition (TTD) in the setting of using both probe vehicle and Bluetooth-based data; (a) travel-time decomposition using probe vehicle data; (b) travel-time decomposition using Bluetooth-based data

the exact device location with a certain known error range, Bluetooth-enabled vehicles are detected passively and location information and reporting depends on the locations of Bluetooth MAC readers. In addition, while instantaneous vehicle speeds can be collected from GPS devices directly, they cannot be measured directly from Bluetooth-based data because only time stamps, detected MACs, and locations of Bluetooth MAC readers are reported. Therefore, the trajectory of the detected device cannot be precisely estimated, and the previously proposed TTD approaches may not be suitable for use with Bluetooth-based data.

At first, the most intuitive solution for addressing the problem of TTD using Bluetooth-based data would seem to be the distance-based method (Neumann 2014; Hellinga et al. 2008). Travel times over multiple links would be decomposed proportionally to link travel times based on link length. This method would be clear and mathematically simple. However, traffic state information would not be integrated into the method, potentially leading to inaccurate results. As stated previously, Gaussian mixture models could be used to model traffic states on roadway segments, and then total travel time could be decomposed by incorporating these traffic states. Probabilistic-based models can incorporate latent variables and better capture the inherent features of traffic data. They are particularly well suited to estimate and predict travel times on signalized arterials when traffic is heterogeneous (Herring et al. 2010; Hofleitner et al. 2012a, b). However, one major drawback to these models is that spatial correlation (also known as the relationship between traffic states on neighboring links) is not considered and each link is treated independently (Hofleitner and Bayen 2011). In reality, because of signal coordination and other factors, links do not operate independently. Upstream traffic states can have an effect on downstream links. Ignoring spatial correlation can lead to an unacceptably large bias. Several previous studies have considered the spatial correlation between links in order to accurately estimate or predict travel times. For example, Min and Wynter (2011) utilized a transient model to mathematically represent the correlation between links. Ramezani and Geroliminis (2012) used Markov chains to model the statistical relationships of link travel-time distributions. They also suggested hidden Markov models as an alternative method to examine link travel-time relationships. Since modeling link relationships is the first step to solving the Bluetooth-based data TTD, the use of a HMM-based model was inspired by the work of Ramezani and Geroliminis (2012).

Based on the above discussion, the intuitive approach was used and three solutions are proposed to solve the Bluetooth-based data TTD problem, including the distance-based approach, GMM-based approach, and HMM-based approach. The distance-based approach was used as the benchmark for comparison, the GMM-based approach was developed to model traffic states on a single link, and the HMM-based approach was developed to incorporate spatial correlation between links. The HMM-based approach can integrate mixed traffic states and transitions between traffic states over successive links. The expectation maximization (EM) algorithm was used to learn model parameters from historical data. One major advantage of the HMM was that it provided the probabilistic relationship between two consecutive links. Following the theme of the proposed machine learning approach, a genetic algorithm (GA) was then used to decompose travel times based on the probabilistic relationship. The proposed HMM approach with GA (HMMGA) was compared with the benchmark approach and the GMM approach with GA (GMMGA). Other optimization models (e.g., chance constrained programming) can also be used to solve the decomposition problem. The benchmark approach simply allocated travel times proportionally based on link lengths. The validation and comparison between the approaches used data sets from real-world Bluetooth sensors.

Methodology

This section focuses primarily on developing the HMMGA. First, the probabilistic models and a corresponding EM algorithm for training the models are described. Next, the adaptation of GAs to solve numerical optimization of TTD with linear constraints is introduced.

Problem Formulation

The goal of TTD is to determine travel times for sublinks given a complete corridor travel time. If the travel times on each sublink are denoted as x_i and the complete travel time spanning over D links is $y.obs$, the summation of each sublink travel time should be equal to the complete travel time

$$\sum_{i=1}^{D} x_i = y.obs \tag{1}$$

The main challenge with TTD is allocating sublink travel times under the linear constraint in Eq. (1). The distributions of travel times over these D successive links can first be characterized by probabilistic models. A well-trained model can be used to make the best guess of travel-time allocation based on a likelihood function. Two probabilistic models, the univariate GMM and HMM, are proposed in this study for comparison. The next step in our TTD approach involves a numerical optimization procedure using GA.

Probabilistic Models

Gaussian Mixture Models

Vehicles may experience different traffic states (free-flow, saturated, or oversaturated) on different links. Therefore, travel times may follow a mixed distribution in which the traffic state is denoted as a latent variable (Guo et al. 2011; Feng et al. 2014). The GMM has already been demonstrated as a potentially useful model for fitting travel times on signalized arterials (Hofleitner and Bayen 2011). The formulation of a GMM is shown in Eq. (2)

$$p(X) = \sum_{\mathrm{k}=1}^{\mathrm{K}} \pi_k \mathrm{N}(X|\mu_k, \Sigma_k) \tag{2}$$

where X = set of historical travel times; k = mixture component number; μ_k = mean travel time of the kth component; Σ_k = covariance matrix; and π_k = mixing coefficient that represents the possibility of each component.

Generally, different numbers of components are set for different links because travel times on different links are modeled independently. However, the GMM does not consider the transition of traffic states over successive links. In reality, traffic states on successive links are spatially correlated. In the GMM, a K-dimensional binary (latent) variable Z is used to indicate to which component an observation belongs. Because of the existence of latent variables, the maximum likelihood estimates cannot be computed directly. The EM algorithm is usually used to estimate the model (Bishop 2006). The basic steps for estimating GMM using EM is described in Algorithm 1. The algorithm will repeatedly apply the E-step and M-step until it reaches convergence.

Algorithm 1. Maximum likelihood estimation of GMMs with EM

Initialize the means μ_k, covariance Σ_k, and evaluate the initial value of log-likelihood.

While The algorithm is not converged **do**

E step: Evaluate the responsibilities using the current parameter values.

$$\gamma(z_{nk}) = \frac{\pi_k \mathrm{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathrm{N}(x_n|\mu_j, \Sigma_j)}$$

M step: Maximize the expected complete data log-likelihood via updating the parameters using the responsibilities at the current step.

$$\begin{aligned} \mu_k^{new} &= \frac{1}{N_k}\sum_{\mathrm{n=1}}^{\mathrm{N}} \gamma(\mathrm{z_{nk}})\mathrm{x_n} \\ \Sigma_k^{new} &= \frac{1}{N_k}\sum_{n=1}^{N} \gamma(\mathrm{z_{nk}})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^{\mathrm{T}} \\ \pi_k^{new} &= \frac{N_k}{N} \\ \text{Where } N_k &= \sum_{n=1}^{N} \gamma(Z_{nk}) \end{aligned}$$

Evaluate the log-likelihood and check the convergence.

$$\ln \, \mathrm{p}(X|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln\left\{\sum_{k=1}^{K} \pi_k \mathrm{N}(x_n|\mu_k, \Sigma_k)\right\}$$

End

Hidden Markov Model

The HMM is a specific instance of the state space model of Bayesian networks in which discrete latent variables are included to represent a specific possible state. The HMM has been widely used for sequential data modeling, including speech recognition and natural language modeling. Traffic states evolving over successive links can also be viewed as a sequential procedure. This sequential procedure can be modeled well using a typical Markov chain: the traffic state (e.g., free flow or congested) on the current link is conditionally dependent on the traffic states of previous links.

Fig. 2 shows the structure of the proposed HMM with a dimension of D variables. The solid circle x_i denotes observed travel times on each link, and the hollow circle Z_i denotes the hidden traffic state. Since the traffic states are not observed directly, they are referred to as latent variables z_i. Each traffic state has a corresponding distribution; given a traffic state z, travel time can be inferred when the corresponding distribution parameters are known. Additionally, traffic may transfer from state j on the previous link (noted as $z_{i-1,j}$) to state k on the next link (noted as $z_{i,k}$). This probability refers to the transition probabilities A_{jk}.

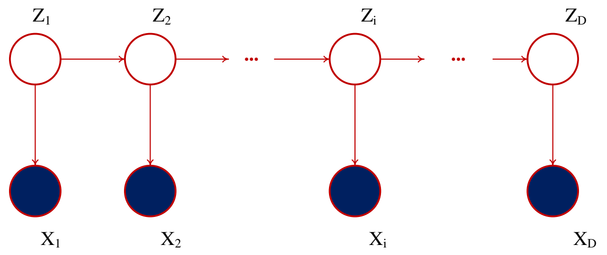

Fig. 2. Hidden Markov model with D variables

A typical HMM is mostly used for modeling state transitions of a dynamic system over the time dimension. Each variable z_i represents the system state of one specific time step i, and the observations are dependent temporarily. However, hidden traffic states are sequentially distributed over space for each observation. Meanwhile, observations are assumed to be independently sampled from the real world. That is, travel time experienced by one vehicle n is independent of that of another vehicle $n + 1$. To handle this specific model structure, modifications were made to a standard HMM.

The derivation of the mathematical procedure for developing and training the HMM is described below. Based on a fundamental knowledge of Bayesian networks (Bishop 2006), basic properties of the HMM can be obtained. The initial latent node z_1 does not have a parent node, and so it has a marginal distribution $\mathrm{p}(z_1)$ represented by a vector of probabilities π with elements $\pi_k \equiv \mathrm{p}(z_{1\ k} = 1)$, represented in Eq. (3)

$$\mathrm{p}(z_1|\pi) = \prod_{k=1}^{K_1} \pi_k^{z_{1k}} \tag{3}$$

where $\sum_k \pi_k = 1$.

The probability distribution of hidden state z_i is dependent on the state of previous latent variable z_{i-1} through a conditional distribution $\mathrm{p}(z_i|z_{i-1})$. Each latent variable is treated as K_i dimensional, where K_i is the number of components for latent variable z_i. Therefore, this conditional distribution corresponds to a table with $K_i \times K_{i+1}$ elements called transition matrix $\boldsymbol{A}$. Each element in the matrix is given by $A_{jk} \equiv \mathrm{p}(z_{ik} = 1|z_{i-1,j} = 1)$, satisfying $0 \leq A_{jk} \leq 1$, with $\sum_k A_{jk} = 1$. The conditional distribution can be written in the form

$$\mathrm{p}(z_i|z_{i-1}, A) = \prod_{k=1}^{K_{i-1}} \prod_{j=1}^{K_i} A_{jk,i-1}{}^{z_{i-1,j} \cdot z_{ik}}, \ \forall\ 2 \leq n \leq D \tag{4}$$

The probabilistic model also needs to specify conditional distributions of observed travel times called the emission probability $\mathrm{p}(x_n|z_n, \phi)$, where ϕ is a set of parameters governing the distribution. In Gaussian distributions, the ϕ here is μ and variance Σ. Given the value of ϕ and observed travel time x_i, the distribution of $\mathrm{p}(x_n|z_n, \phi)$ consists of a vector of K_i numbers corresponding to the K_i possible states of binary variable z_i. The emission probability is represented in the following form:

$$\mathrm{p}(x_n|z_n, \phi) = \prod_{k=1}^{K_i} \mathrm{p}(x_i|\phi_k)^{z_{ik}}, \ \forall\ 1 \leq i \leq D \tag{5}$$

The proposed HMM is a heterogeneous model in which the number of states K for each latent variable and transition probabilities for each two adjacent segments are different. The emission probability for travel times on each segment also varies. When training the HMM, these model parameters need to be learned from historical data.

Using the d-separation criterion for directed Bayesian networks (Bishop 2006), the joint probability distribution over both the latent and observed variables is given by

$$\mathrm{p}(X, Z|\theta) = \mathrm{p}(z_1|\pi)\left[\prod_{i=2}^{D} \mathrm{p}(z_i|z_{i-1}, A)\right]\left[\prod_{m=1}^{D} \mathrm{p}(x_m|z_m, \phi)\right] \tag{6}$$

The likelihood function for the HMM is obtained based on the above joint distribution by marginalizing the latent variables

$$\mathrm{p}(X|\theta) = \sum_{Z} \mathrm{p}(X, Z|\theta) \tag{7}$$

In order to obtain optimal model parameters, this likelihood function is maximized by changing parameters. By taking a logarithm over the likelihood function, the summation over Z occurs inside the logarithm. This results in complicated expressions with no closed-form solution. Instead, working on the complete data set in which the corresponding values of the latent variable Z are given is much more practical. Since it is assumed that the observations are independent, the observations can be factorized over number of observations N. The complete data likelihood is thus given by the form

$$\mathrm{p}(X,Z|\theta)=\prod_{n=1}^{N}\Bigg\{\prod_{k=1}^{K}\pi_k^{z_{1k}}\cdot\Bigg[\prod_{i=2}^{D}\Bigg(\prod_{k=1}^{K}\prod_{j=1}^{K}A_{jk,i-1}^{z_{i-1,j}\cdot z_{ik}}\Bigg)\Bigg]\cdot\prod_{m=1}^{D}\Bigg(\prod_{k=1}^{K}\mathrm{p}(x_m|\phi_k)^{z_{mk}}\Bigg)\Bigg\}\tag{8}$$

As this is explicitly factorized over latent variables, applying a logarithm makes the maximization much more straightforward:

$$\ln\mathrm{p}(X,Z|\theta)=\sum_{n=1}^{N}\sum_{k=1}^{K_1}z_{1k}\cdot\ln\pi_k+\sum_{n=1}^{N}\sum_{i=2}^{D}\sum_{k=1}^{K_{i-1}}\sum_{j=1}^{K_i}z_{(i-1,j)}z_{ik}\ln A_{jk,i-1}+\sum_{n=1}^{N}\sum_{i=1}^{D}\sum_{k=1}^{K_i}z_{ik}\cdot\ln\mathrm{p}(x_i|\phi_{ik})\tag{9}$$

In practice, the latent variables are not given, so only the expectation of complete data log-likelihood can be used. The expectation is given by

$$\begin{aligned}\mathrm{Q}(\theta,\theta^{old})&=\sum_{Z}\mathrm{p}(Z|X,\theta^{old})\cdot\ln\mathrm{p}(X,Z|\theta)\\&=\sum_{Z}\gamma(Z)\cdot\ln\mathrm{p}(X,Z|\theta)\\&=\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(Z_{1\ k})\cdot\ln\pi_k\\&\quad+\sum_{n=1}^{N}\sum_{i=2}^{D}\sum_{k=1}^{K}\sum_{j=1}^{K}\xi(Z_{i-1,j},Z_{ik})\ln A_{jk,i-1}\\&\quad+\sum_{n=1}^{N}\sum_{i=1}^{D}\sum_{k=1}^{K}\gamma(Z_{ik})\cdot\ln\mathrm{p}(x_i|\phi_{ik})\end{aligned}\tag{10}$$

where $\gamma(z_i)=\mathrm{p}(z_i|X,\theta^{old})$ = marginal posterior distribution of latent variable z_i and $\xi(Z_{i-1},Z_i)=\mathrm{p}(z_{i-1},z_i|X,\theta^{old})$ = joint posterior distribution of two successive latent variables.

The EM algorithm can be used to efficiently find the maximization solution. In this E step, the algorithm takes the initial model parameters θ^{old} and evaluates the posterior distribution of latent variables $\mathrm{p}(Z|X,\theta^{old})$, which here refers to the quantities $\gamma(z_i)$ and $\xi(Z_{i-1},Z_i)$. In the M step, it then uses these posterior distributions to maximize $\mathrm{Q}(\theta,\theta^{old})$ with respect to $\{\pi,A,\Phi\}$. This can be implemented by using appropriate Lagrange multipliers and taking the first-order derivative. The derived results are shown in Algorithm 2.

The evaluation of quantities $\gamma(z_i)$ and $\xi(Z_{i-1},Z_i)$ can be implemented by the forward-backward algorithm (Rabiner 1989), also known as the α-β algorithm. In the proposed HMM, the quantities $\alpha(z_i)$ and $\beta(z_i)$ can be evaluated recursively using the following equation at each E step:

$$\begin{aligned}\alpha(z_i)&=\begin{cases}\mathrm{p}(x_i|z_i)\sum\limits_{z_{i-1}}\alpha(z_{i-1})\cdot p(z_i|z_{i-1}), & i\geq 2\\ \prod_{k=1}^{K}\{\pi_k\cdot\mathrm{p}(x_1|\phi_{1\ k})\}^{z_{1\ k}}, & i=1\end{cases},\\ \beta(z_i)&=\begin{cases}\sum\limits_{z_{i+1}}\beta(z_{i+1})\cdot\mathrm{p}(x_{i+1}|z_{i+1})\cdot\mathrm{p}(z_{i+1}|z_i), & i<D\\ 1, & i=D\end{cases}\end{aligned}\tag{11}$$

Next, $\gamma(z_i)$ and $\xi(Z_{i-1},Z_i)$ are evaluated using the following equations. Notice the existence of joint probability $\mathrm{p}(X)$ occurs only at the denominator, so it will be canceled out in the M step

$$\gamma(Z_i)=\mathrm{p}(z_i|X)=\frac{\mathrm{p}(X|Z_i)\mathrm{p}(Z_i)}{\mathrm{p}(X)}=\frac{\alpha(Z_i)\cdot\beta(Z_i)}{\mathrm{p}(X)}\tag{12}$$

$$\xi(Z_{i-1},Z_i)=\frac{\alpha(z_{i-1})\cdot\mathrm{p}(x_i|z_i)\cdot\mathrm{p}(z_i|z_{i-1})\cdot\beta(z_i)}{\mathrm{p}(X)}\tag{13}$$

Before making use of the forward-backward algorithm in practice, Bishop (2006) noted that α and β should be normalized to avoid the problem of arithmetic underflow. This is because both these values can go to zero at an exponential rate. The normalization is obtained by introducing a scaling factor $c_i=\mathrm{p}(x_i|x_1,x_2,\ldots,x_{i-1})$ and $\mathrm{p}(x_1,x_2,\ldots,x_i)=\prod_{\mathrm{m=1}}^{\mathrm{i}}c_m$. The normalized versions of α and β are given by

$$\hat{\alpha}(z_i)=\frac{\alpha(z_i)}{\mathrm{p}(x_1,\ldots,x_i)}=\frac{\alpha(z_i)}{\prod_{m=1}^{i}c_m},\qquad\hat{\beta}(z_i)=\frac{\beta(z_i)}{(\prod_{m=i+1}^{D}c_m)}\tag{14}$$

The corresponding recursive equation for evaluating normalized α, β, and log-likelihood is given in Algorithm 2.

Algorithm 2. Maximum Likelihood Estimation of HMM with EM

Initialize the π, means μ, variance Σ, and transition matrix A; then evaluate the initial value of log-likelihood.

While The algorithm is not converged **do**

E step: Evaluate the marginal posterior distribution of latent variable (γ) and joint posterior distribution of two successive latent variables (ξ).

1. Apply forward recursive equation to evaluate $\hat{\alpha}(z_i)$

$$c_i\cdot\alpha(z_i)=\mathrm{p}(x_i|z_i)\cdot\sum_{z_{i-1}}\alpha(z_{i-1})\cdot\mathrm{p}(z_i|z_{i-1})$$

2. Evaluate and store the scaling factor c_i

$$c_i=\sum_{j=1}^{K}\mathrm{p}(x_i|z_i)\cdot\sum_{z_{i-1}}\hat{\alpha}(z_{i-1})\cdot\mathrm{p}(z_i|z_{i-1})$$

3. Apply a backward recursive equation and use scaling factor to evaluate $\hat{\beta}(z_i)$

$$c_{i+1}\cdot\beta(z_i)=\sum_{z_{i+1}}\beta(z_{i+1})\cdot\mathrm{p}(x_{i+1}|z_{i+1})\cdot\mathrm{p}(z_{i+1}|z_i)$$

4. Evaluate $\gamma(z_i)$ and $\xi(Z_{i-1}, Z_i)$

$$\gamma(z_i) = \hat{\alpha}(z_i) \cdot \hat{\beta}(z_i)$$

$$\xi(Z_{i-1}, Z_i) = \frac{\hat{\alpha}(z_{i-1}) \cdot \mathrm{p}(x_i|z_i) \cdot \mathrm{p}(z_i|z_{i-1}) \cdot \hat{\beta}(z_i)}{c_i}$$

M step: Maximize the expected complete data log-likelihood via updating the parameters using γ and ξ at the current step.

$$\mu_{ik} = \frac{\sum_{n=1}^{N} \gamma(z_{ink}) \cdot X_{in}}{\sum_{n=1}^{N} \gamma(z_{ink})},$$
$$\Sigma_{ik} = \frac{\sum_{n=1}^{N} \gamma(z_{ink}) \cdot (x_{in} - \mu_{ik}) \cdot (x_{in} - \mu_{ik})^T}{\sum_{n=1}^{N} \gamma(z_{ink})}$$
$$\pi_k = \frac{\sum_{n=1}^{N} \gamma(z_{ink})}{\sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{ink})},$$
$$A_{jk,i-1} = \frac{\sum_{n=1}^{N} \xi(z_{i-1,j,n}, z_{i,k,n})}{\sum_{n=1}^{N} \sum_{k=1}^{K} \xi(z_{i-1,j,n}, z_{i,k,n})}$$

Evaluate the log-likelihood and check the convergence.

$$\ln \mathrm{p}(X) = \sum_{n=1}^{N} \sum_{i=1}^{D} \ln c_i$$

End

Once the model parameters are learned from historical data, it is important to find the most probable sequence of hidden states given a new observation and its corresponding maximum likelihood. The maximum likelihood will be used as a basis to allocate travel times in the numerical optimization. The most probable sequence is obtained by evaluating the maximum likelihood over all the possible sequences for hidden states. When the dimension of variables (i.e., D) is large, the computation cost can increase exponentially. In practice, this can be efficiently solved using the Viterbi algorithm in linear time (Viterbi 1967). This is implemented by first transferring the directed Bayesian network (Fig. 2) to a factor graph (Fig. 3).

Next, node z_D is treated as a root and passes the message from leaf nodes (z_i) to the root. The messages passed from node to factor and factor to node are given by

$$\mu_{\mathrm{Z_i}} \rightarrow \mathrm{f}_{(\mathrm{i+1})}(Z_i) = \mu_{\mathrm{f_i}} \rightarrow \mathrm{z_i}(\mathrm{Z_i})$$
$$\mu_{f_{i+1}} \rightarrow z_{i+1}(Z_{i+1}) = \max_{z_i}(\ln \mathrm{f}_{\mathrm{i+1}}(z_i, z_{i+1}) + \mu_{z_i \rightarrow \mathrm{f}_{\mathrm{i+1}}}(z_i)) \qquad (15)$$

where $\mu_{Z_i} \rightarrow f_{i+1}(Z_i)$ = message passed from node z_i to factor f_{i+1} and $\mu_{f_{i+1}} \rightarrow z_{i+1}(Z_{i+1})$ = message passed from factor f_{i+1} to node z_{i+1}.

A further recursive equation can be obtained by eliminating $\mu_{Z_i} \rightarrow f_{i+1}(Z_i)$. By introducing $\omega(z_i) \equiv \mu_{z_i \rightarrow f_{i+1}}(z_i)$, a recursive equation for calculating $\omega(z_D)$ can be obtained, which can be proven to be the maximum likelihood for this given observation.

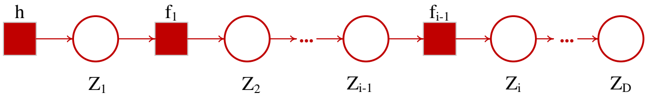

Fig. 3. Factor graph of hidden Markov model with D variables

The most probable sequence can be obtained in the message passing procedure by recording which hidden state of previous segments gives the maximum likelihood to the next segment. Once the hidden state of the root by maximization over z_D is obtained, the algorithm can then backtrack to the previous hidden states. This algorithm is described in Algorithm 3.

Algorithm 3. Viterbi Algorithm for Finding Most Probable Sequence

```
While i <= D do
  1. Recursively estimate $\omega(z_i)$.
     $\omega(z_1) = \ln \mathrm{p}(z_1) + \ln \mathrm{p}(x_1|z_1)$
     $\omega(z_{i+1}) = \ln \mathrm{p}(x_{i+1}|z_{i+1}) + \max_{z_i}\{\ln \mathrm{p}(z_{i+1}|z_i) + \omega(z_i)\}$
  2. Store each of the previous hidden states $(z_i)$ that results in the maximum $\omega(z_{i+1})$.
  $\psi(k_i) = \arg\max \omega(z_i)$ where $k \in \{1, \ldots, K(i)\}$
End
Evaluate the maximum likelihood $\max_{z_D} \omega(z_D)$ and use the function $\psi(k_i)$ to backtrack previous hidden states
     $k_{n-1}^{\max} = \psi(k_n^{\max})$
```

Numerical Optimization

Once the probabilistic models are well developed and the appropriate parameters are known, travel times on sublinks can be allocated based on the maximum likelihood from these models. The only constraint is that the summation of allocated travel times must equal the observed complete travel time for the corridor. This can be formulated as an optimization problem and the objective function set as the negative maximum likelihood:

$$\begin{aligned}&Minimize - LogLikelihood\\&\text{subject to } 0 \leq x_i < y.obs, \quad \forall\ i \in [1, D-1]\\&\sum_{i=1}^{D-1} x_i < y.obs\end{aligned} \qquad (16)$$

where $y.obs$ = observed travel time on complete successive links.

The objective function is given by the negative log-likelihood function. It can be estimated directly for the GMM by summing over all possible traffic states. The log-likelihood for the HMM can be obtained via Algorithm 3. In the optimization problem, the degree of original constraint D can be reduced to $D - 1$ by transferring the equality constraint to the inequality constraint. This step can help to explicitly define the lower and upper bounds of decision variables.

The GA is a widely used artificial intelligence method for solving numerical optimization problems. It ensures that the near global optimal solution is found and will not converge merely on a local optimal solution. GAs are also very efficient and flexible at adapting to specific problems. Essentially, a GA (Janikow and Michalewicz 1990) searches the solution domain via two genetic operators known as crossover (crossover between the parents to form new offspring) and mutation (mutate new offspring at certain genes of the chromosome). The feasible solutions (represented by chromosomes) are evaluated based on a fitness function $f(x)$ to find the optimal solution. However, the standard GA operators must be modified to handle specific constraints. Michalewicz and Janikow (1993) proposed a method to handle linear constraints for GAs by dynamically calculating the boundaries for GA operators. Following their method, the standard crossover and mutation operators were modified.

Two types of mutations are used: uniform and boundary mutation. Suppose the two selected parent chromosomes at the tth iteration are $s_v^t = (v_1, v_2, \ldots, v_m)$ and $s_w^t = (w_1, w_2, \ldots, w_m)$. The algorithm selects a random gene v_k (element of a feasible solution) of the chromosome and then dynamically calculates its lower and upper boundaries $l_{(k)}^{s_v^t}$ and $u_{(k)}^{s_v^t}$ based on the set of constraints. Next, if uniform mutation is applied, a random value from $v_k' \in [l_{(k)}^{s_v^t}, u_{(k)}^{s_v^t}]$ is drawn uniformly. If boundary mutation is applied, the selected gene is simply set as $v_k' = l_{(k)}^{s_v^t}$ or $u_{(k)}^{s_v^t}$. Both uniform and boundary mutations have a preset chance of being applied.

The crossover operator was implemented in three ways: single crossover, simple crossover, and whole crossover. The crossovers exchanged the information between the two parent chromosomes for either one single gene, a sequence of partial chromosome, or the entire chromosome by a scaling parameter of a.

The single crossover exchanged the selected genes v_k and w_k from the parent chromosomes in the form

$$\begin{aligned} v_k' &= \mathrm{a} \cdot w_k + (1 - \mathrm{a}) \cdot v_k \\ w_k' &= \mathrm{a} \cdot v_k + (1 - \mathrm{a}) \cdot w_k \end{aligned} \tag{17}$$

The value of a can be randomly sampled according to

$$a = \begin{cases} \left[\max\left(\dfrac{l_k^{s_w} - w_k}{v_k - w_k}, \dfrac{u_k^{s_v} - v_k}{w_k - v_k}\right), \min\left(\dfrac{l_k^{s_v} - v_k}{w_k - v_k}, \dfrac{u_k^{s_w} - w_k}{v_k - w_k}\right)\right], & \text{if } v_k > w_k \\ [0, 0], & \text{if } v_k = w_k \\ \left[\max\left(\dfrac{l_k^{s_v} - v_k}{w_k - v_k}, \dfrac{u_k^{s_w} - w_k}{v_k - w_k}\right), \min\left(\dfrac{l_k^{s_w} - w_k}{v_k - w_k}, \dfrac{u_k^{s_v} - v_k}{w_k - v_k}\right)\right], & \text{if } v_k > w_k \end{cases} \tag{18}$$

The simple crossover exchanges a partial sequence of chromosome in the form

$$\begin{aligned} s_v^{t+1} &= (v_1, v_2, \ldots, v_k, w_{k+1} \cdot a + v_{k+1} \cdot (1 - a), \ldots, w_{D-1} \cdot a + v_{D-1} \cdot (1 - a)) \\ s_w^{t+1} &= (w_1, w_2, \ldots, w_k, v_{k+1} \cdot a + w_{k+1} \cdot (1 - a), \ldots, v_{D-1} \cdot a + w_{D-1} \cdot (1 - a)) \end{aligned} \tag{19}$$

The value $a \in [0,1]$ determines how much information will be exchanged. If a equals 0, then nothing changes. If a equals 1, the two parent chromosomes were flipped. For a certain value a, a simple crossover may generate solutions that exceed the constrained domain. Therefore, the algorithm needs to find the maximum value of a that keeps the crossed chromosomes in the constrained domain.

The whole crossover simply exchanges the two parent chromosomes by the value of $a \in [0,1]$. It can be shown that this operation will always limit the crossed chromosomes in the constrained domain (Michalewicz and Janikow 1993)

$$\begin{aligned} s_v^{t+1} &= \mathrm{a} \cdot s_w^t + (1 - \mathrm{a}) \cdot s_v^t \\ s_w^{t+1} &= \mathrm{a} \cdot s_v^t + (1 - \mathrm{a}) \cdot s_w^t \end{aligned} \tag{20}$$

The convergence of the GA is checked by evaluating the maximum likelihood function, which is a negative of the objective function. Once it reaches convergence, the near-optimal allocation solution is obtained.

Experiments

A case study using the two Bluetooth-based data sets was used to compare the three model approaches. The results from the validation are then discussed to evaluate the performance of the three approaches.

Bluetooth-Based Travel-Time Experiment Setup

Previously, the Smart Transportation Lab at the University of Arizona had developed and installed Bluetooth-based traffic sensors to collect travel-time data on several major arterials in Tucson, including Speedway Boulevard. Speedway was used as the experimental arterial in this case study. Fig. 4 shows the main section of Speedway from Euclid Avenue to Kolb Road. In this experiment, two subsections were defined: (1) three links on west Speedway from Euclid to Campbell Avenue and (2) six links on east Speedway from Alvernon Way to Kolb. These two subsections had very different traffic characteristics. The shorter, western subsection had much shorter signal spacing and carried more traffic. The eastern subsection had longer and more evenly distributed signal spacing. In general, traffic on west Speedway was more heterogeneous.

Table 1 shows detailed information for both Speedway subsections. Travel-time data were collected along Speedway using the Bluetooth sensors. Two data sets consisting of complete trajectory observations were collected for model training and validation. These data sets were collected from May 12 to June 9, 2015, for west Speedway and from June 10 to July 25, 2014, for east Speedway. These two data sets are denoted as west

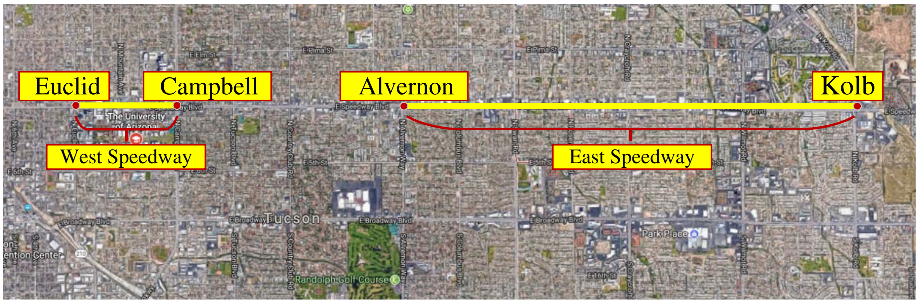

Fig. 4. Experimental arterial in Tucson, Arizona (Speedway) (map data © 2017 Google)

Table 1. Configuration of the Experimental Arterial: Speedway

Link identifier	Cross streets	Length [m (mi)]	Free-flow speed [km/h (mph)]	Free-flow travel time (s)	Speed limit [km/h (mph)]
West Speedway					
Link 1	Euclid-Park	259 (0.1610)	59.35 (36.88)	15.72	56.3 (35)
Link 2	Park-Mountain	407 (0.2530)	62.19 (38.64)	23.57	56.3 (35)
Link 3	Mountain-Campbell	784 (0.4870)	61.59 (38.27)	45.70	56.3 (35)
East Speedway					
Link 4	Alvernon-Columbus	813 (0.5054)	63.92 (39.72)	45.81	56.3 (35)
Link 5	Columbus-Swan	803 (0.4992)	63.89 (39.70)	45.26	56.3 (35)
Link 6	Swan-Rosemont	805 (0.5004)	63.90 (39.71)	45.37	56.3 (35)
Link 7	Rosemon-Craycroft	797 (0.4954)	63.87 (39.69)	44.93	56.3 (35)
Link 8	Craycroft-Wilmot	1,657 (1.0300)	65.92 (40.96)	90.52	56.3 (35)
Link 9	Wilmot-Kolb	1,592 (0.9895)	69.73 (43.33)	82.20	64.4 (40)

Speedway (with 4,422 observations) and east Speedway (with 5,860 observations).

Results

The three TTD approaches described in the methodology section (the HMMGA, GMMGA, and benchmark) were tested using the case study data sets described above (west Speedway and east Speedway). The benchmark approach simply allocated travel times proportionally based on the link lengths. The probabilistic models with GA were implemented using the R programming package. To achieve reliable results, a 10-group cross validation was applied to each of these three approaches. Specifically, the data sets were divided into 10 groups. Nine groups were used for model training and one group was used for validation. This procedure was repeated 10 times so that each group was used once for validation. The following three error measures were used to evaluate the effectiveness of the algorithms:

$$\text{Mean Absolute Error (MAE)} = \frac{1}{N}\sum_{i=1}^{N}|tt_{decmp}^{(i)} - tt_{act}^{(i)}| \tag{21}$$

$$\text{Mean Absolute Percentage Error (MAPE)}$$
$$= \frac{1}{N}\sum_{i=1}^{N}\left|\frac{tt_{decmp}^{(i)} - tt_{act}^{(i)}}{tt_{act}^{(i)}}\right| \tag{22}$$

$$\text{Root Mean Square Error (RMSE)} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(tt_{decmp}^{(i)} - tt_{act}^{(i)})^2} \tag{23}$$

where $tt_{decmp}^{(i)}$ and $tt_{act}^{(i)}$ = decomposed and actual travel times, respectively.

Figs. 5(a and b) summarize the mean and standard deviation of the MAEs, MAPEs, and RMSEs for the three approaches based on the 10-group cross validation.

Figs. 5(a and b) show that, overall, the HMMGA outperformed the benchmark and GMMGA algorithms with both data sets. It had much lower MAE, MAPE, and RMSE values. The superior performance of the HMMGA was especially notable on the west Speedway subsection, where traffic was more heterogeneous and mixed. This was because the HMMGA not only modeled the travel-time distributions but also considered the transitions between traffic states over successive links. The benchmark approach had lower MAE and RMSE values than the GMMGA on the east Speedway subsection. This was possibly because the traffic on east Speedway was more consistent because of large and evenly distributed signal spacing. In situations like this, the benefit of a mixed distribution structure is minor. Lastly, MAPE was a more consistent and intuitive error measure compared to MAE and RMSE. The HMMGA MAPE was up to 72% lower compared to the benchmark approach.

Additionally, the effectiveness of the algorithms at the link level is worth examining. The examination focused on using MAPE to identify differences among the three approaches. Figs. 6(a and b) show the MAPE distributions for each link of the west and east Speedway subsections. Some specific findings are summarized below:

1. For west Speedway, the average MAPEs of the HMMGA were generally lower than those of the benchmark approach and the GMMGA, and the standard deviations of MAPEs of HMMGA were smaller than those of the benchmark approach and the GMMGA. However, the average MAPEs of the HMMGA were only slightly lower than those in Links 1 and 3.

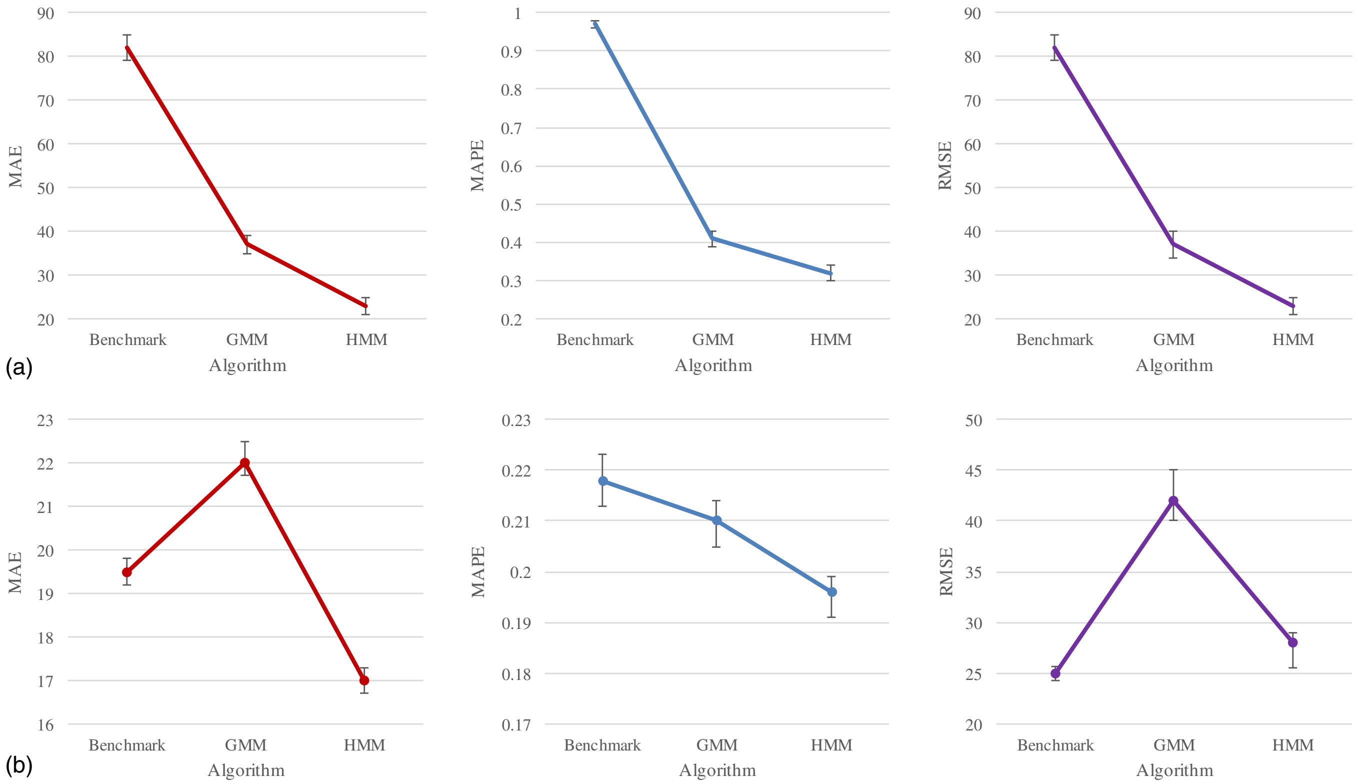

Fig. 5. Summary of error measures for the three algorithms: (a) west Speedway; (b) east Speedway

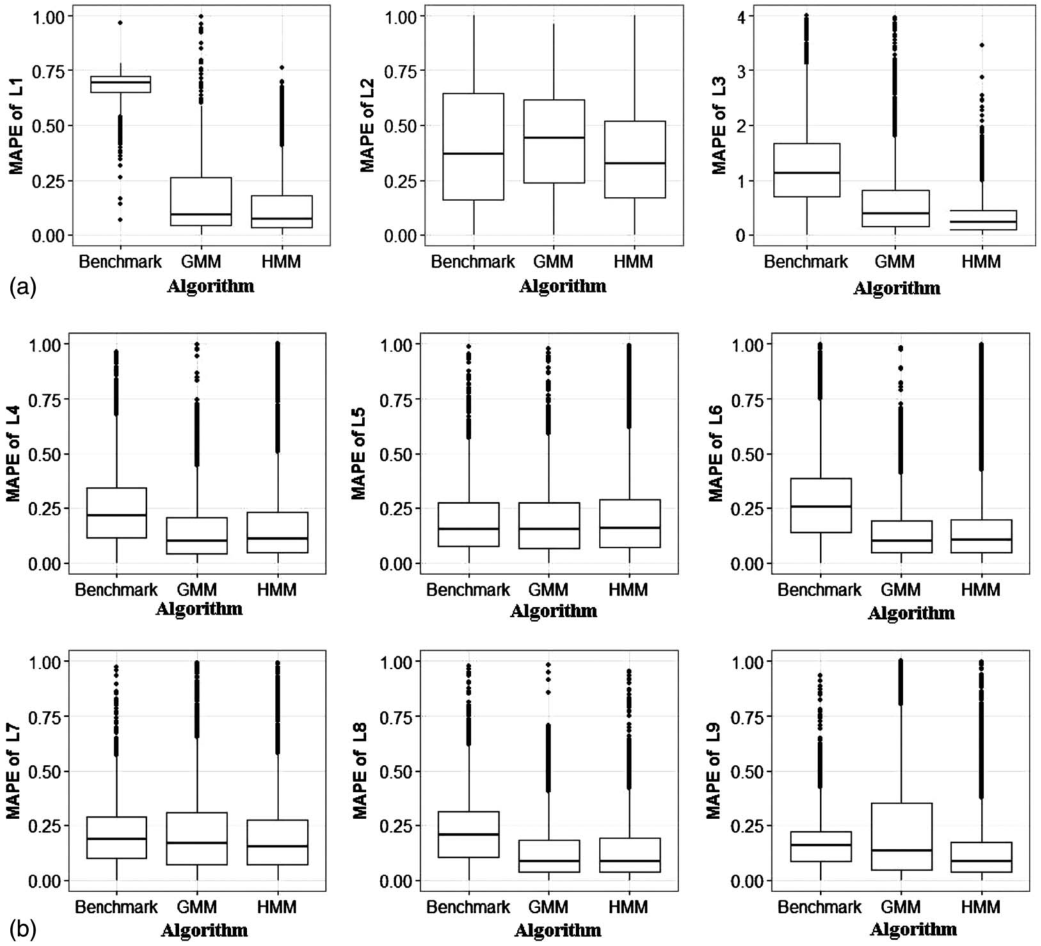

Fig. 6. Distribution of MAPE by link: (a) west Speedway; (b) east Speedway

2. For east Speedway, both the averages and standard deviations of the HMMGA MAPEs were similar to the GMMGA MAPEs. Even the differences of the MAPE average and standard deviation values among the benchmark approach, GMMGA, and HMMGA were insignificant. The benefits of using the HMMGA were more significant on the links with heterogeneous and mixed traffic. For example, Links 1, 3, 4, 6, and 8 carried mixed traffic because they intersected major north-south arterials. These larger intersections were near major trip production and attraction sites, including shopping malls.
3. The MAPE standard deviations of Link 2 were noticeably wider than others. This may be due to pedestrian calls. Link 2 passes through the University of Arizona, with classroom buildings located on both sides of Speedway between Park Avenue and Mountain Avenue. Many students cross Speedway to access these buildings, using the pedestrian push buttons and affecting signal timing. The resulting signal timing uncertainty due to pedestrian calls may have caused the larger standard deviations.

Overall, the HMMGA outperformed the benchmark and GMMGA approaches, especially when traffic was heterogeneous. Moreover, error comparisons showed that travel times on successive links are indeed spatially dependent. Previous decomposition algorithms (Hellinga et al. 2008; Hofleitner and Bayen 2011) that ignored this dependency might lead to bias. Also, if travel times on certain individual links are collected, the HMMGA can be further improved by incorporating these values as observed nodes in the sequential Markov chain (Fig. 2), thus increasing TTD accuracy. The benchmark and GMMGA approaches without the Markov chain cannot incorporate individual link travel times in TTD.

Conclusions

The problem of missing data, especially MNAR, is very common and presents many challenges in Bluetooth-based travel-time data and other data sets. This study proposed a machine learning–based approach to decomposing corridor travel times into link travel times. Three models were proposed, including a benchmark using a distance-based model, a Gaussian mixture model, and a hidden Markov model. The GMM was used to model the travel-time distributions on a single link, while the HMM was used to incorporate spatial correlation and model the distributions over multiple links. After modeling the travel-time distributions on a single link or a segment, genetic algorithms were used to find the optimal numeric solutions to allocate total observed travel times to single links. The distance-based model served as the benchmark model and was compared to both the GMMGA and HMMGA models. Although this study only illustrated the application of the HMMGA and GMMGA approach in completing Bluetooth-based travel-time sets with MNAR, the HMMGA and GMMGA also could easily be applied to other types of travel-time data (e.g., probe vehicle data) wherever accurate TTD is desired. As long as the input data are in the same format as Bluetooth-based data, the HMMGA and GMMGA can be used.

Validation results using a study corridor along Speedway Boulevard in Tucson, Arizona, showed that the HMMGA outperformed the other two approaches (the GMMGA and the benchmark approach), especially when traffic was heterogeneous. The average MAPE was up to 72% lower compared to the benchmark approach. Unlike the GMMGA, the HMMGA can incorporate spatial dependency as a sequential transition procedure over successive links. This helped reduce decomposition errors even when traffic was heterogeneous and mixed.

The current HMMGA approach requires a complete travel-time data set large enough to train the model. Future research should consider improving model training with incomplete travel-time data sets and testing more spatial or even spatial-temporal models to investigate link correlations.

Acknowledgments

The authors would like to thank the Regional Transportation Authority, the Pima Association of Governments, and the City of Tucson for their funding support. We would like to thank Paul Casertano, Michael Hicks, Bob Hunt, and Francisco Leyva for providing valuable advice and technical support in this project. We are grateful to Simon Ramos at the Arizona Department of Transportation for initial installation of the Bluetooth sensors. Special thanks go to Payton Cooke for his proofreading assistance.

References

Bhaskar, A., and Chung, E. (2013). “Fundamental understanding on the use of Bluetooth scanner as a complementary transport data.” *Transp. Res. Part C: Emerging Technol.*, 37, 42–72.

Bishop, C. (2006). *Pattern recognition and machine learning (information science and statistics)*, Springer, New York.

Day, C., et al. (2010). “Evaluation of arterial signal coordination: Methodologies for visualizing high-resolution event data and measuring travel time.” *Transp. Res. Rec.*, 2192, 37–49.

Feng, Y., Hourdos, J., and Davis, G. A. (2014). “Probe vehicle based real-time traffic monitoring on urban roadways.” *Transp. Res. Part C: Emerging Technol.*, 40, 160–178.

Guo, F., Rakha, H., and Park, S. (2010). “Multistate model for travel time reliability.” *Transp. Res. Rec.*, 2188, 46–54.

Haghani, A., Hamedi, M., Sadabadi, K. F., Young, S., and Tarnoff, P. (2010). “Data collection of freeway travel time ground truth with Bluetooth sensors.” *Transp. Res. Rec.*, 2160, 60–68.

Hellinga, B., Izadpanah, P., Takada, H., and Fu, L. (2008). “Decomposing travel times measured by probe-based traffic monitoring systems to individual road segments.” *Transp. Res. Part C: Emerging Technol.*, 16(6), 768–782.

Herring, R., Hofleitner, A., Abbeel, P., and Bayen, A. (2010). “Estimating arterial traffic conditions using sparse probe data.” *Proc., Int. IEEE Conf. on Intelligent Transportation Systems*, IEEE, Piscataway, NJ, 929–936.

Hofleitner, A., and Bayen, A. (2011). “Optimal decomposition of travel times measured by probe vehicles using a statistical traffic flow model.” *Proc., Int. IEEE Conf. on Intelligent Transportation Systems*, IEEE, Piscataway, NJ, 815–821.

Hofleitner, A., Herring, R., Abbeel, P., and Bayen, A. (2012a). “Learning the dynamics of arterial traffic from probe data using a dynamic Bayesian network.” *Intell. Transp. Syst. IEEE Trans.*, 13(4), 1679–1693.

Hofleitner, A., Herring, R., and Bayen, A. (2012b). “Arterial travel time forecast with streaming data: A hybrid approach of flow modeling and machine learning.” *Transp. Res. Part B: Methodol.*, 46(9), 1097–1122.

Janikow, C. Z., and Michalewicz, Z. (1990). “A specialized genetic algorithm for numerical optimization problems.” *Proc., 2nd Int. IEEE Conf. on Tools for Artificial Intelligence*, IEEE, Piscataway, NJ, 798–804.

Jenelius, E., and Koutsopoulos, H. N. (2015). “Probe vehicle data sampled by time or space: Consistent travel time allocation and estimation.” *Transp. Res. Part B: Methodol.*, 71, 120–137.

Michalewicz, Z., and Janikow, C. Z. (1993). “Handling constraints in genetic algorithms.” *Proc., 4th Int. Conf. on Genetic Algorithms*, Morgan Kaufmann Publishers, San Francisco.

Min, W., and Wynter, L. (2011). “Real-time road traffic prediction with spatio-temporal correlations.” *Transp. Res. Part C: Emerging Technol.*, 19(4), 606–616.

Mohammad, A., and Ashish, K. (2013). “Travel time prediction on signalised urban arterials by applying SARIMA modelling on Bluetooth data.”

Proc., 36th Australasian Transport Research Forum, Queensland Univ. of Technology, Brisbane, QLD, Australia.

Neumann, T. (2014). “Accuracy of distance-based travel time decomposition in probe vehicle systems.” *J. Adv. Transp.*, 48(8), 1087–1106.

Quayle, S., Koonce, P., DePencier, D., and Bullock, D. (2010). “Arterial performance measures with media access control readers: Portland, Oregon, pilot study.” *Transp. Res. Rec.*, 2192, 185–193.

Rabiner, L. R. (1989). “A tutorial on hidden Markov models and selected applications in speech recognition.” *Proc. IEEE*, 77(2), 257–286.

Ramezani, M., and Geroliminis, N. (2012). “On the estimation of arterial route travel time distribution with Markov chains.” *Transp. Res. Part B: Methodol.*, 46(10), 1576–1590.

Schafer, J. L. (1997). *Analysis of incomplete multivariate data*, CRC Press, Boca Raton, FL.

Viterbi, A. J. (1967). “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm.” *IEEE Trans. Inf. Theory*, 13(2), 260–269.

Yang, S., An, C., Wu, Y. J., and Xia, J. (2017). “Origin-destination based travel time reliability.” *Transp. Res. Rec.*, 2643, 139–159.

Yang, S., and Wu, Y. J. (2016). “Mixture models for fitting freeway travel time distributions and measuring travel time reliability.” *Transp. Res. Rec.*, 2594, 95–106.

Zheng, F., and van Zuylen, H. (2013). “Urban link travel time estimation based on sparse probe vehicle data.” *Transp. Res. Part C: Emerging Technol.*, 31(111), 145–157.