Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Transit smart card data mining for passenger origin information extraction*

Xiao-lei MA[†1], Yin-hai WANG[†‡1], Feng CHEN[2], Jian-feng LIU[2]

([1]*Department of Civil and Environmental Engineering, University of Washington, Seattle, WA 98195-2700, USA*)

([2]*Beijing Transportation Research Center, Beijing 100073, China*)

[†]E-mail: {xiaolm, yinhai}@uw.edu

Received Feb. 23, 2012; Revision accepted June 11, 2012; Crosschecked Aug. 20, 2012

Abstract: The automated fare collection (AFC) system, also known as the transit smart card (SC) system, has gained more and more popularity among transit agencies worldwide. Compared with the conventional manual fare collection system, an AFC system has its inherent advantages in low labor cost and high efficiency for fare collection and transaction data archival. Although it is possible to collect highly valuable data from transit SC transactions, substantial efforts and methodologies are needed for extracting such data because most AFC systems are not initially designed for data collection. This is true especially for the Beijing AFC system, where a passenger's boarding stop (origin) on a flat-rate bus is not recorded on the check-in scan. To extract passengers' origin data from recorded SC transaction information, a Markov chain based Bayesian decision tree algorithm is developed in this study. Using the time invariance property of the Markov chain, the algorithm is further optimized and simplified to have a linear computational complexity. This algorithm is verified with transit vehicles equipped with global positioning system (GPS) data loggers. Our verification results demonstrated that the proposed algorithm is effective in extracting transit passengers' origin information from SC transactions with a relatively high accuracy. Such transit origin data are highly valuable for transit system planning and route optimization.

Key words: Transit smart card, Automated fare collection (AFC), Bayesian decision tree, Markov chain, Origin inference
doi:10.1631/jzus.C12a0049 **Document code:** A **CLC number:** U121; TP391

1 Introduction

US Energy Information Administration (2007) stated that "more than 50% of commuters drive their own cars to work". In China, more and more travelers commute by transit. For example, the percentage of public transit (including rail transit) riders in Beijing increased from 36.8% in 2008 to 38.9% in 2009 (Beijing Transportation Research Center (BTRC), 2010a). This implies that traffic congestion in metropolitan areas can be mitigated if public transit services take a larger share of commuting trips. However, a commuter's choice depends on the utility associated with each available mode. Transit service must be improved to increase its utility and therefore attract more riders.

Transit passenger origin-destination (OD) data are crucial for transit system planning and route optimization (Li, 2009). Collecting such OD data, however, is extremely difficult and expensive using traditional paper-survey-based approaches (Hofmann *et al*., 2009; Reddy *et al*., 2009). Automated fare collection (AFC) systems contain rich spatial and temporal information through contactless smart cards (SCs), each with a unique ID, which significantly reduces manpower to collect transit passenger OD data. However, most AFC systems are not designed for OD data collection (Pelletier *et al*., 2011); hence, further data processing and analysis is necessary for passenger information extraction (Barry *et al*., 2009). In this paper we present a Bayesian decision tree

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 51138003) and the Beijing Transportation Research Center (BTRC), China

based statistical approach to infer the passenger origin from the imperfect SC transaction data, which is the first step for the transit OD estimation technique.

The remainder of this paper is organized as follows. First, the potential problem with the Beijing AFC system is fully described. Then, we briefly discuss relevant studies in transit OD estimation methods for entry-only AFC systems. This is followed by a novel Markov chain based Bayesian decision tree algorithm which is proposed to infer passenger origin data based on the existing smart data characteristics. This algorithm is verified in the next section using global positioning system (GPS) data with a detailed result analysis. Conclusions of the study are made at the end of this paper.

2 Problem statement

In May 10, 2006, the Beijing Transportation Corporation started to issue the Beijing Transportation SC to transit riders. If a user pays the transit fare with the SC, up to 60% discount can be received. Such a large discount quickly stimulated the use of SCs. In 2010, more than 90% of the transit users paid for their transit trips with their SCs (BTRC, 2010b). There are a total of 16 million SC transactions every day. Among these transactions, 52% are from flat-rate bus riders. This implies that the OD information for flat-rate bus riders is essential to form the complete OD matrix of Beijing transit riders. All SC transactions are archived in a database at the BTRC, China. The high market penetration rate and tremendous daily transactions ensure a great data source, and present challenges for the transit rider OD extraction as well.

The transit rider OD matrix can potentially be extracted from the SC transaction database. However, this is not a straightforward task. Two major challenges must be addressed to obtain good quality OD data. The challenges originate from the design of the SC scan system for the flat-rate buses. Since passengers pay a fixed rate to the flat-rate buses, only check-in scan is considered necessary in SC scan system design (Zhao *et al*., 2007). Compared to the distance-based fare bus riders, flat-rate bus users do not have check-out records. This creates the first challenge in the OD extraction: where does a passenger get off a flat-rate bus? Furthermore, the scan system does not save the location or direction information on check-in scans and this creates the second challenge: where does a passenger get on a flat-rate bus?

The two challenges induce two very interesting research topics: (1) how to identify the transit stop ID for a check-in scan and (2) at which transit stop does the passenger get off the flat-rate bus? Given the fixed route of transit vehicles, known distance between stops, and transaction records stored in the database, including SC ID, route number, driver ID, transaction time, remaining balance, and transaction amount, it is possible to estimate a flat-rate bus user's check-in and check-out stops through data mining and data fusion techniques. However, the accuracy of the extracted OD data depends largely on the quality of the data processing algorithms (Zhang, 2002).

Many cities in China employ the SC system for transit services. Almost all the systems suffer from the same problem as the Beijing system. A solution for passenger boarding and alighting information extraction will be beneficial to most transit agencies in China. This paper focuses on the first challenge to identify the transit stop ID for a check-in scan. A Markov chain based Bayesian decision tree algorithm is developed to resolve this problem.

3 Related works

Many OD matrix inference approaches have been investigated over the past years. Research on Metropolitan Transit Authority (MTA)'s MetroCard system in New York City (Barry *et al*., 2002; 2009) revealed the feasibility of station-to-station OD matrix generation in the entry-only AFC subway system. Zhao *et al*. (2007) and Rahbee (2009) proposed a transit OD matrix estimation algorithm for origin-only AFC data from the Chicago Transit Authority rail system. However, their algorithms primarily focus on the rail system, where boarding at fixed stations is easier to locate than in bus transit systems. Pelletier *et al*. (2010) undertook a thorough literature review on transit SC data usage, and concluded that properly processing SC data can enhance the strategic, tactical, and operational performances for transit agencies. Trépanier *et al*. (2007; 2009) conducted several studies on the AFC system in the National Capital Region of Canada, and developed algorithms

to extract travel information from SC transaction data for transit performance measures. They evaluated various transit statistics and demonstrated the feasibility of developing a transit performance measurement system using SC data. Most of the aforementioned studies are based on the entry-only AFC system, where boarding information is known in advance. In several existing AFC systems with missing boarding stops, researchers incorporated other data sources to jointly infer boarding locations, such as automated passenger counter (APC) data, schedule data, and GPS data. Farzin (2008) outlined a process to construct an automated transit OD matrix based on SC and GPS data in Brazil. Nassir *et al*. (2011) integrated APC data, GPS data, and transit schedule data with AFC data to estimate the stop-level passenger origin and destination. The AFC system of the city of Changchun (China) lacks both boarding and alighting stops; hence, Zhang *et al*. (2007) designed an on-bus questionnaire to match each passenger's boarding time for origin inference. To the best of our knowledge, few studies were undertaken to infer passenger's origin from the entry-only AFC system with the missing boarding information. Review of the existing literature did not identify any approach suitable for passenger OD information extraction from Beijing SC transaction data. Hence, an algorithm applicable for the Beijing AFC system is highly desired.

4 Methodology

As mentioned in Section 2, the boarding stop and bus direction information is missing in the Beijing transit SC transaction database. Boarding stop is not directly available from the database. However, most passengers scan their cards immediately when boarding and almost all passengers have completed the check-in scan before arriving at the next stop. This indicates that the first passenger's transaction time can be safely assumed as the group of passengers' boarding time at the same stop. The challenge is then to identify the bus location at the moment of the SC transaction, so that we can infer the onboard stop for that passenger. However, this is not easy as the SC system for the flat-rate bus does not record the bus location. We know the time each transaction occurred on a bus of a particular route under the operation of a particular driver, but nothing else is known from the SC transaction database. Nonetheless, we can extract boarding volume changes with time and passengers who made transfers. By mining these data and combining transit route maps, we may accomplish our goal. Therefore, a two-step approach is designed for passenger origin data extraction: SC data clustering and transit stop recognition. Details of each step are described below.

4.1 Smart card data clustering

4.1.1 Transaction data classification

First, we need to sort SC transactions by the transit vehicle number. This results in a list of SC transactions in the vehicle for the entire period of operations for each day. During the operational period, the vehicle may have two to ten round-trip runs depending on the round-trip length and roadway condition. At a terminal station, a transit vehicle may take a break or continue running, so there is no obvious signal for the end of a trip (a trip is defined as the journey from one terminus to the other terminus). Meanwhile, there are a varying number of passengers at each stop, including some stops with no passengers.

For stops with several passengers boarding, all transactions can be classified into one group based on the interval between their transactions. Thus, the clustered SC transactions can be represented by a time series of check-in passenger volumes at stops (Table 1).

Table 1 The four clustered smart card transactions

Parameter	Value/Description			
	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Stop ID	Unknown	Unknown	Unknown	Unknown
Stop name	Unknown	Unknown	Unknown	Unknown
Total number of transactions	18	9	11	27
Transaction timestamp	5:26:36	5:41:02	5:44:18	5:48:53
Time difference	0:14:26	0:03:16	0:04:35	0:01:00

In Table 1, the 'total number of transactions' indicates the total boarding passengers in one stop, 'transaction timestamp' is recorded as the time when the first passenger boards in this stop, and 'time difference' means the elapsed time between the boarding times at this stop and at the next stop with boarding passengers. Unlike most entry-only AFC systems in

the US, the stop name and ID from each transaction are unknown in the Beijing AFC system. Most buses in service follow the predefined order of stops; however, it is still possible that there is no passenger boarding in a specific stop, and thus two consecutive SC transaction clusters do not necessarily correspond to two physically consecutive stops. Obviously, this further complicates the situation and the algorithm should map each cluster into the corresponding boarding stop ID.

In summary, the SC data clustering algorithm contains the following three steps:

1. All transaction data for each bus are sorted by the transaction timestamp in ascending order.

2. For two consecutive records, if their transaction time difference is within 60 s, these two transactions are included in one cluster; otherwise, another cluster is initiated.

3. If the transaction time difference for two consecutive records is greater than 30 min or driver changing occurs, it is likely that the bus has arrived at the terminus, and for this bus, one bus trip has completed. The next record will be the beginning for the next bus trip.

The results of the clustering process are several sequences of clustered transactions. Each sequence may contain one or more trips of the transit vehicle. For particular routes, due to the limited space in the terminus or busy transit schedule, bus layover time may be too short to be used as a separation symbol for trips. Such buses may have a very long clustered sequence, which makes the pattern discovery process very challenging. Furthermore, unfamiliar passengers or passengers boarding from the check-out doors (this occurs for very crowded buses) may take longer than 60 s to scan their cards. The delayed transaction may cause cluster assignment errors. Again, this adds extra challenge to the follow-up passenger origin extraction process.

4.1.2 Transaction cluster sequence segmentation

Beijing has a huge transit network with nearly 1000 routes. It is quite common to see passengers transfer between transit routes. Through transfer activity analysis, we can further segment the clustered transaction sequence into shorter series to reduce the uncertainty in passenger OD estimation (Jang, 2010). The key principle used for transfer stop identification is that the alighting stop in the previous route is assumed to be spatially the closest to the boarding stop for the next route. This is reasonable because most passengers choose the closest stop for transit transfer (Chu and Chapleau, 2008).

Assume a passenger k makes a transfer from route i to route j. If either of the two routes is distance-based-rate bus or a subway line, then we can identify the name of the transfer station. Even if both routes are flat-rate bus routes, for a unique transferring location, we can still use the transfer information to identify the transfer bus stop ID and name. In addition, the walking distance between the two stops is needed to infer the time when the flat-rate bus arrives at the transfer stop.

Based on the identified transfer stops, we can further segment the transaction cluster sequence into shorter cluster series. Each series is bounded by either the termini or the identified bus stops. The segmented series of transaction clusters will be used as the input for the subsequent transit stop inference algorithm.

4.2 Data mining for transit stop recognition

If we treat each segmented series of transaction clusters as an unknown pattern, this unknown pattern can be considered as a sample of the sequential stops on the bus route. If every stop has boarding passengers, this unknown pattern is identical to the known bus stop sequence. Also, since distance and speed limit between stops are known, travel time between stops is highly predictable if there is no traffic jam. In reality, however, there may be a varying distribution of passengers boarding at any given stop, and roadway congestion may result in unpredictable delays. Therefore, unknown pattern recognition is a very challenging issue. Once the unknown pattern is recognized, the boarding stop for any passenger becomes clear.

The Bayesian decision tree algorithm is one of the widely used data mining techniques for pattern recognition (Janssens *et al*., 2006). Each node in the Bayesian decision tree is connected through the Bayesian conditional probability, and the entire tree is constructed directionally from the root node to the leaf nodes. Applying this technique to the current problem, we can represent the known starting stop as the root. Denoting the current boarding stop ID at time step k as S_k, and the next boarding stop ID at time step k+1 as S_{k+1}, according to Bayesian inference theory (Bayes and Price, 1763), S_{k+1} can be calculated as

$$S_{k+1} = \arg\max_j \Pr(S_{k+1} = j \mid S_1, S_2, ..., S_k), \quad (1)$$

where $\Pr(S_{k+1} \mid S_1, S_2, ..., S_k)$ is the conditional probability of the next boarding stop being S_{k+1}, given the previous boarding stop sequence S_1, S_2, …, S_k.

A Bayesian decision tree represents many possible known patterns. We need to compute the probability for each known pattern to match the unknown pattern. By further observation, we can find that the probability of passengers boarding at S_{k+1} at time step k+1 is related only to whether the last boarding stop is S_k at time step k. This is because if the transaction time and corresponding bus location for SC transaction cluster k are known, the next SC transaction cluster k+1 relies only on how fast the bus travels during the time period between SC transaction clusters k and k+1. In this case, an SC transaction series can be recognized as a Markov chain process. Markov chain is a stochastic process with the property that the next state relies only on the current state. Therefore, S_{k+1} can be rewritten as

$$\begin{aligned} S_{k+1} &= \arg\max_j \Pr(S_{k+1} = j \mid S_1, S_2, ..., S_k) \\ &= \arg\max_j \Pr(S_{k+1} = j \mid S_k = i) \\ &\text{s.t. } i<j. \end{aligned} \quad (2)$$

The single-step Markov transition probability is defined as $\Pr(S_{k+1}=j|S_k=i)$, also denoted as p_{ij}, with i and j being the stop IDs. Without losing generality, we assume the bus is moving outbound with an increasing trend of stop ID toward the destination. Then the transition probability matrix $\boldsymbol{\Pi}$ can be simplified as

$$\begin{aligned} \boldsymbol{\Pi} &= \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & & \vdots \\ p_{(n-1)1} & p_{(n-1)2} & \cdots & p_{(n-1)n} \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{pmatrix} \\ &= \begin{pmatrix} 1-\sum_{i=2}^{n} p_{1i} & p_{12} & \cdots & p_{1n} \\ 0 & 1-\sum_{i=2}^{n} p_{2i} & \cdots & p_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & p_{(n-1)n} \\ 0 & 0 & \cdots & 1 \end{pmatrix}, \end{aligned} \quad (3)$$

where n is the total number of stops for the bus route. This transition probability matrix plays a vital role in determining the potential stop ID for the next time step.

4.2.1 Transition matrix generation

To recognize the unknown pattern, it is critical to develop a measure to quantify p_{ij}, the possibility of the next boarding stop being stop j conditioned on the previous boarding stop being i. The higher the p_{ij}, the more likely the next SC transaction cluster corresponds to boarding passengers at stop j. In other words, p_{ij} represents the probability of the next SC transaction cluster timestamp being the bus boarding time at stop j. That is to say, the boarding time in stop j for cluster k+1 can be predicted based on the travel distance from stops i to j and average bus speed. Then, the calculated time can be used as an indicator to compare with the real transaction timestamp for cluster k+1. From this point, the average speed between stops i and j will be a key variable. If the timestamp is t_k for cluster k, and t_{k+1} for cluster k+1, then the bus travel time from time steps k to k+1 is $t_{k+1}-t_k$. The stop distance between stops j and i is D_{ij}. Then the average bus travel speed V_{ij} can be expressed as

$$V_{ij} = \frac{D_{ij}}{t_{k+1} - t_k}, \quad (4)$$

where V_{ij} is a random variable depending on the traffic condition at the moment. V_{ij} is considered to be normally distributed, and its probability density function can be adopted to quantify p_{ij}.

In the speed normal distribution, the mean travel speed μ_{ij} and standard deviation σ_{ij} can be calculated from all buses with GPS devices in the same route. Under this circumstance, the boarding time for each stop can be inferred by matching GPS data and stop location information. Using the inferred boarding time difference and distance between stops i and j, we can calculate the mean travel speed μ_{ij} and standard deviation σ_{ij} as a priori information. Note that the mean speed and standard deviation are not dependent on GPS data, but can be obtained using other data sources such as distance-based-rate SC transaction data. A sensitivity analysis further demonstrates the algorithm's robustness even with fluctuations on both mean speed and standard deviation.

Then, the transition probability can be reformulated as

$$\begin{aligned} p_{ij} &= \Pr(S_{k+1}=j \mid S_k=i) \\ &= \int_{z_{ij}-\Delta}^{z_{ij}+\Delta} \frac{1}{\sqrt{2\pi}} \exp(-z^2/2)\mathrm{d}z \\ &\approx \frac{1}{\sqrt{2\pi}} \exp(-z_{ij}^2/2)\cdot 2\Delta, \end{aligned} \tag{5}$$

where $z_{ij}=(V_{ij}-\mu_{ij})/\sigma_{ij}$ is the standardized travel speed between stops j and i. Δ is a small increment of the travel speed, and will not impact the algorithm result, since it is a common term for each transition probability.

Each element in the transition matrix can be quantified in the same way as Eq. (5). With the complete transition matrix, the unknown pattern of SC transaction series can be recognized as

$$\begin{aligned} &[S_{k+1}, S_k, S_{k-1}, ..., S_1] \\ &= \arg\max_{S_1 S_2 ... S_{k+1}} \Pr(S_{k+1}, S_k, S_{k-1}, ..., S_1) \\ &= \arg\max_{S_1 S_2 ... S_{k+1}} \left(\Pr(S_{k+1} \mid S_k, S_{k-1}, ..., S_1)\Pr(S_k, S_{k-1}, ..., S_1)\right) \\ &= \arg\max_{S_1 S_2 ... S_{k+1}} \left(\Pr(S_{k+1} \mid S_k)\Pr(S_k \mid S_{k-1})\cdots\Pr(S_2 \mid S_1)\right) \\ &= \arg\max_{S_1 S_2 ... S_{k+1}} \left(\prod_{n=1}^{k} \Pr(S_{n+1}=j \mid S_n=i)\right) \\ &= \arg\max_{S_1 S_2 ... S_{k+1}} \left(\sqrt[k+1]{\prod_{n=1}^{k} \Pr(S_{n+1}=j \mid S_n=i)}\right) \\ &\triangleq \arg\max_{S_1 S_2 ... S_{k+1}} \overline{P}(k+1), \end{aligned} \tag{6}$$

where $\overline{P}(k+1)$ denotes the geometric mean of the probabilities of the passenger boarding stop sequence at time step k+1. It is also the probability that the identified stop sequence matches the unknown pattern.

Although GPS data are used, as a matter of fact, the algorithm is not very sensitive to either the mean speed or standard deviation, as will be demonstrated later by variable sensitivity analysis.

5 Algorithm implementation and optimization

5.1 Implementation

Fig. 1 illustrates the process of using the Markov chain based Bayesian decision tree algorithm for transit origin information extraction. Before estimating a passenger boarding stop, we need to assume that there must be passengers boarding within the first three stops in one route, and this stop is used as the root node in the Bayesian decision tree to initiate the algorithm. This assumption is reasonable for the Beijing transit system because for approximately 95% flat-rate buses, there are transactions within the first three stops, at any time period, based on our survey.

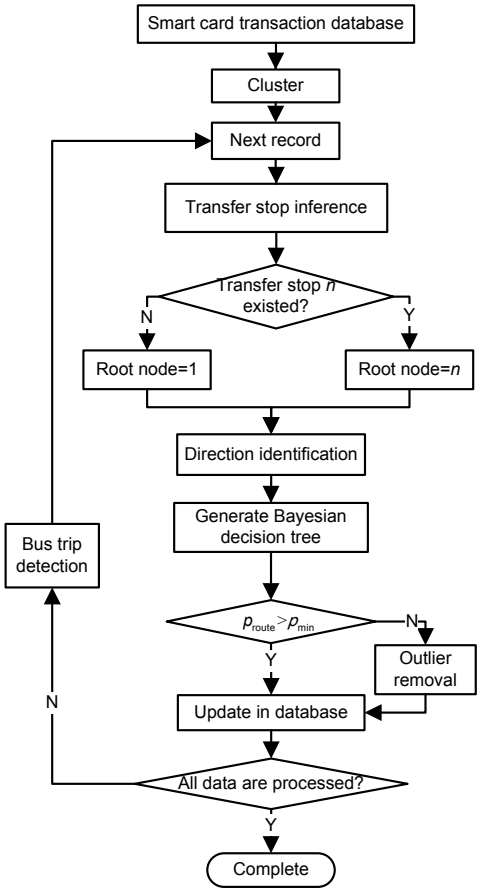


Fig. 1 Flow chart of the Markov chain based Bayesian decision tree algorithm

p_{route}: Bayesian decision path probability; p_{min}: path probability threshold

As mentioned in previous sections, due to the nature of transaction data, several defects need to be addressed in the process of the Markov chain based Bayesian decision tree algorithm.

1. Direction identification

The Beijing transit AFC system does not log the travel direction information for each route. We need to determine whether the bus is traveling inbound or outbound before algorithm execution. The solution is

that we construct two Bayesian decision trees in each direction, with the roots being the first stop (terminus), second stop, and third stop, respectively. Then the probability of the most likely stop sequence from each of the six trees will be compared and the one with the highest path probability wins.

2. Outlier removal

As mentioned in Section 4.1, in some cases, the delayed transactions impact the accuracy of the clustering algorithm, and these abnormal transactions are also labeled as outliers. The principal difficulty is that two inconsistent SC transactions by timestamp that should be classified in one cluster may be read separately, and thus the latter will be classified as another cluster for the next stop. For instance, at a particular stop, one passenger gets on the bus and pays the fare at 8:00 AM, and another passenger swipes his/her SC at 8:10 AM. Due to the relatively large transaction timestamp gap, the second transaction will be assigned to another cluster. In this case, the boarding stop ID will be misidentified.

The strategy used to remove these outliers is that we set a path probability threshold $p_{\min}$. If the decision tree starts with a false root node (e.g., the initial SC transaction cluster is misidentified), its error will be accumulated and extended in the following leave nodes. As a result, the path probability decreases as the time step increases, and the initial cluster is then abandoned; thus, we restart the computation process from the next cluster, until we find the path probability that exceeds the threshold.

3. Bus trip detection

The journey from the initial bus stop to the terminus is defined as a bus trip. The bus terminus is designed for bus turning, layover, and driver change. It is also the starting stop on the bus timetable. However, in the Beijing transit network, some bus termini are located in the busy street or have limited space. Hence, buses using these termini have to begin their next trip in a short time period without causing an obstruction. This is a challenging issue in the procedure of passenger origin inference, since the initial stop (root node) in the Bayesian decision tree may be misidentified if the bus trip is mistakenly detected. Similar to the solution to outlier removal, the same threshold $p_{\min}$ can ensure the correct initial root node in the Bayesian decision tree.

This algorithm was successfully implemented in Microsoft Visual C#, and all the SC transaction data were stored in the Microsoft SQL server database.

5.2 Computational performance optimization

Although we have illustrated the mathematical form of the Markov chain based Bayesian decision tree in theory, this algorithm presented above has not been applied in the real dataset. Cooper (1990) has proven that the Bayesian decision tree algorithm is an NP-hard problem, which means this algorithm cannot be solved in polynomial time. The conventional approach to calculating the path probability for all the potential boarding stop sequences is computationally expensive, especially for the long sequences. To better explain this challenge, an example is shown in Fig. 2.

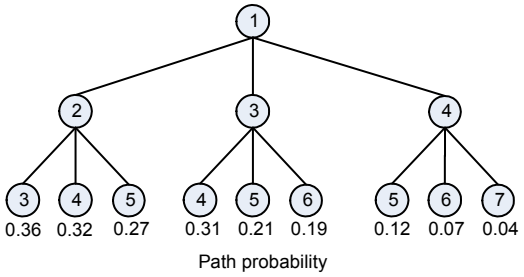


Fig. 2 A Bayesian decision tree algorithm example

Assume the initial boarding stop is 1. The potential stop in the next step could be stop 2, stop 3, or stop 4, since they are all in the reachable range. Assuming the situations are similar for the remaining stops, a decision tree is fully established following the approach as shown in Fig. 2. The traditional exhaustive search is to traverse each potential path and select the maximum probability. Based on this method, we need to calculate the path probability nine times. This implies that the number of paths to be calculated increases exponentially as the time step increases. However, at time step 3, there are two or more paths ending with stops 3, 4, and 5. Before carrying out the computation in the next time step, we can compare the probabilities of the paths with the same ending stop, and choose the maximum one, which is also called the 'partial best path'.

In time step 3, only the following five paths are selected: 1→2→3, 1→2→4, 1→2→5, 1→3→6, and 1→4→7. Recall that in the Markov chain model, given a previous state sequence, the probability of the

current state depends only on the previous state. Hence, the most probable paths in time step 4 are guaranteed by the five paths calculated in time step 3 without extra computations of other paths. According to Eq. (6), we can express the optimized procedure as

$$\overline{P}(k+1)=\max_{i,j}\left(\overline{P}(k)\sqrt[k+1]{\Pr(S_{k+1}=j\mid S_k=i)}\right). \quad (7)$$

We can now calculate the probability at each time step recursively until the end of the route. Computing the probability in this way is far less computationally expensive than calculating the probabilities for all sequences. Denoting the total number of stops for a specific route as n, and classifying the SC transactions in m clusters, which correspond to m time steps in Bayesian decision trees, the computational complexity of the exhaustive approach is $O(m^n)$. While using the optimized algorithm, the computational complexity is only $O(mn)$. With the optimization, the algorithm can be solved in finite time, and can be efficiently applied in reality.

6 Validation and analysis

By installing GPS receivers on flat-rate buses, we can collect the bus geospatial information and spot speed data in a real-time manner. These data provide the opportunity to validate the Markov chain based Bayesian decision tree algorithm developed in this study for passenger origin data extraction. GPS coordinates and timestamp can be used to determine bus boarding and alighting location and time. First, the geographical features of bus stops and consecutive GPS records for each bus are joined using latitude and longitude coordinates. Then, by matching the passenger check-in time in the SC transaction database, the boarding stop ID can be associated with each transaction. The inferred stop ID using GPS data can be considered as 'ground truth' data for comparison.

In this section, the Markov chain based Bayesian decision tree algorithm is first validated using GPS data for two flat-rate based routes, and then a bus travel speed sensitivity analysis is conducted to investigate the impacts of the speed mean and standard deviation calculated from GPS data.

6.1 Algorithm validation

Two flat-rate based routes with buses equipped with GPS devices were selected (Fig. 3). The SC transaction data and GPS data were recorded on April 7, 2010. Route 300 is a loop route with 30 stops. Thereby, the initial stop and terminus for Route 300 are identical. There are totally 32 stops in Route 67 for both inbound and outbound directions.

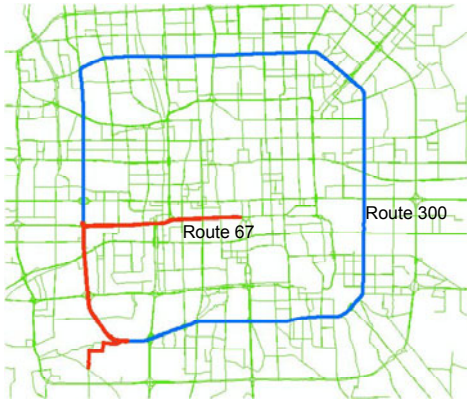


Fig. 3 Map of Routes 67 and 300 in the Beijing transit network (the red line is Route 67 and the blue line is Route 300)
References to color refer to the online version of this figure

The algorithm results are listed in Tables 2 and 3. There are a total of 12187 SC transactions mapped with GPS data for Route 67, and 24059 transactions for Route 300. The error is defined as the stop ID difference (two stops next to each other should have consecutive IDs) between the ground truth stop based on GPS data and the inferred stop using the proposed algorithm, the 'percentage in results' is equal to the number of records divided by the number of inferred transactions, and the 'percentage in total transactions' is calculated as the number of records divided by the number of total transactions. For Route 67, 94% passenger boarding stops were deducted by the algorithm. There are 8055 recognized boarding stops whose absolute deviations to the true stop IDs are within three, accounting for approximately 70% of the total identified stops or 66% of the entire SC transactions. However, for Route 300, the algorithm result accuracy decreases slightly compared with that of Route 67. Out of the identified stops, 68% are within three stops from the ground truth, corresponding to 65% of the entire SC transactions. The reduced accuracy is largely due to the loop route for

Route 300, whose buses start and end in the same place and bus drivers have only a very short layover time in the terminus. This makes it very difficult to detect the initial stop. A wrong root stop will impact the algorithm results consequently.

Table 2 Results of the Markov chain Bayesian decision tree algorithm for Route 67

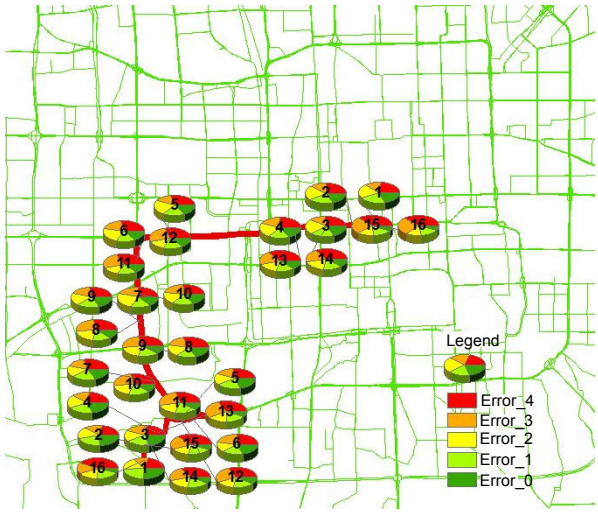
Stop ID error	Number of records	Percentage in results	Percentage in total transactions
<1	1529	13%	13%
<2	3926	34%	33%
<3	5882	51%	48%
<4	8055	70%	66%
≥4	3380	30%	28%
Total	11435	100%	94%

Table 3 Results of the Markov chain Bayesian decision tree algorithm for Route 300

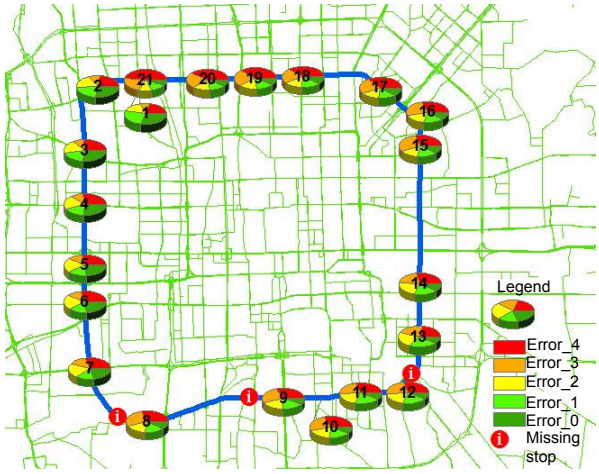
Stop ID error	Number of records	Percentage in results	Percentage in total transactions
<1	3035	14%	13%
<2	7232	33%	30%
<3	10988	50%	48%
<4	14961	68%	65%
≥4	6904	32%	28%
Total	21865	100%	93%

To further illustrate the effectiveness of the proposed algorithm, the error distribution for each stop is spatially presented on a geographic information system (GIS) map (Figs. 4a and 4b). For Route 67, errors for most stops are evenly distributed at different levels; however, the last few stop errors suffer from a slight increase. This is because the errors generated in previous stops are propagated to the last few stops and augmented. For Route 300, there are three missing stops (shown as exclamation marks on the GIS map) which cannot be detected by the proposed algorithm. Although the missing stops incur the erroneous identification of subsequent stops, the proposed algorithm is still able to limit the stop misdetection to within an acceptable range. The proposed algorithm generates a relatively high accuracy, and demonstrates its robust error tolerance capability.

The results are quite encouraging even with incomplete GIS information. In the Beijing transit network, the errors within three stops are acceptable for transit planning level study, since these stops are affiliated mostly with the same traffic analysis zone (TAZ).



(a)



(b)

Fig. 4 Stop-level error distribution maps for Route 67 (a) and Route 300 (b) (numbers represent stop order)

6.2 Travel speed sensitivity analysis

Recall that in computing the transition matrix, mean travel speed μ and standard deviation σ were calculated from GPS data. However, there are still many flat-rate routes without GPS devices. To understand how the algorithm result changes when the mean travel speed and standard deviation are

inaccurate, a sensitivity analysis was carried out. Tables 4 and 5 show the results when mean travel speed and standard deviation change for Route 67, where $\Delta\sigma$ is defined as the standard deviation change, and $\Delta\mu$ is defined as the mean change of speed.

Table 4 Results of the Markov chain Bayesian decision tree algorithm for Route 67 when the travel speed standard deviation ($\Delta\sigma$) changes

Stop ID error	Number of records				
	$\Delta\sigma$=0	$\Delta\sigma$=3	$\Delta\sigma$=−3	$\Delta\sigma$=5	$\Delta\sigma$=−5
<1	1529	1555	1407	1498	1453
<2	3926	4025	3804	3960	3834
<3	5882	5837	5887	5824	5824
<4	8055	8234	8099	8114	7928
≥4	3380	3128	3432	3486	3389
Total	11435	11362	11531	11600	11317

Table 5 Results of the Markov chain Bayesian decision tree algorithm for Route 67 when the mean travel speed ($\Delta\mu$, in km/h) changes

Stop ID error	Number of records				
	$\Delta\mu$=0	$\Delta\mu$=3	$\Delta\mu$=−3	$\Delta\mu$=5	$\Delta\mu$=−5
<1	1529	1387	1423	1238	1302
<2	3926	3479	3653	3660	3503
<3	5882	5487	5622	5491	5294
<4	8055	7878	7967	7714	7334
≥4	3380	3611	3405	4272	4087
Total	11435	11489	11372	11986	11421

Tables 4 and 5 show that the impacts of the travel speed standard deviation changes are less than those of the mean travel speed changes. However, the accuracy of the algorithm is not highly sensitive to either the travel speed standard deviation or the mean. This is not surprising, because in a normal distribution, mean and standard deviation influence only the shape of the probability density function. As long as we make a reasonable assumption for bus travel speed calculation, the algorithm results will not fluctuate significantly. Even if there are no available GPS data, we can still calculate the travel speed, either from other flat-fare routes with GPS devices or from distance-based fare routes, which share common stops with the 'no-GPS' flat-fare route. Alternatively, traffic detectors can be used to calculate the travel speed.

7 Conclusions

Different from most entry-only AFC systems in other countries, the Beijing AFC system does not record boarding location information when passengers get on the buses and swipe their SCs. This creates challenges for passenger OD estimation.

This study aims to tackle this issue. With further investigations on SC transactions data, we propose a Markov chain based Bayesian decision tree algorithm to infer passengers boarding stops. This algorithm is based on Bayesian conditional probability theory, and the probability density function of travel speed normal distribution is used to measure the randomness of passenger boarding stops, where its mean and variance are not sensitive to algorithm accuracy and thereby not dependent on other data sources. Moreover, we can use the time invariance of the Markov chain model to further reduce the computational complexity of the algorithm to linear load. The effectiveness of the optimized algorithm is proven using the SC transaction data from two routes.

This algorithm can still be improved in many ways. For instance, the algorithm does not perform well when the travel speed is not distinct; i.e., the travel speed probability calculated for each stop is similar. The countermeasure for this issue is to incorporate more feature matrices; e.g., the closeness between each stop and the subway stations or tourist spots is a factor to quantify the attractiveness for the transit ridership.

In summary, the Markov chain based Bayesian decision tree algorithm provides an effective data mining approach for passenger origin data extraction. It offers a start for mining transit passenger ODs from the SC transaction data for transit system planning and operations. The long-term research goal of this study is to establish a network-wide transit OD matrix for the Beijing transit performance measurement program, which can be used to estimate trip demand and optimize the bus schedule and route for policy makers and transit operators. This research contributes an effective statistical approach to extracting boarding information for the widely used entry-missing AFC system in China, which is feasible for implementation by practitioners. Furthermore, the algorithm sheds light on other unknown sequential pattern recognition problems in transportation, and can be generalized to other similar issues.

References

Barry, J.J., Newhouser, R., Rahbee, A., Sayeda, S., 2002. Origin and destination estimation in New York City with automated fare system data. *Transp. Res. Rec.*, **1817**: 183-187. [doi:10.3141/1817-24]

Barry, J.J., Freimer, R., Slavin, H., 2009. Use of entry-only automatic fare collection data to estimate linked transit trips in New York City. *Transp. Res. Rec.*, **2112**:53-61. [doi:10.3141/2112-07]

Bayes, T., Price, R., 1763. An essay towards solving a problem in the doctrine of chances. *Phil. Trans. R. Soc. Lond.*, **53**:370-418. [doi:10.1098/rstl.1763.0053]

BTRC (Beijing Transportation Research Center), 2010a. Beijing Transport Annual Report 2010. Available from http://www.bjtrc.org.cn/InfoCenter%5CNewsAttach%5C%5C3891f531-3019-4d28-9b70-29c58217b50d.pdf (in Chinese) [Accessed on Aug. 23, 2011].

BTRC (Beijing Transportation Research Center), 2010b. Beijing Transportation Smart Card Usage Survey. Research Report, unpublished (in Chinese).

Chu, K.K.A., Chapleau, R., 2008. Enriching archived smart card transaction data for transit demand modeling. *Transp. Res. Rec.*, **2063**:63-72. [doi:10.3141/2063-08]

Cooper, G.F., 1990. The computational complexity of probabilistic inference using Bayesian belief networks. *Artif. Intell.*, **42**(2-3):393-405. [doi:10.1016/0004-3702(90)90060-D]

Farzin, J.M., 2008. Constructing an automated bus origin-destination matrix using farecard and global positioning system data in Sao Paulo, Brazil. *Transp. Res. Rec.*, **2072**:30-37. [doi:10.3141/2072-04]

Hofmann, M., Wilson, S., White, P., 2009. Automated Identification of Linked Trips at Trip Level Using Electronic Fare Collection Data. 88th Annual Meeting of Transportation Research Board, p.18.

Jang, W., 2010. Travel time and transfer analysis using transit smart card data. *Transp. Res. Rec.*, **2144**:142-149. [doi:10.3141/2144-16]

Janssens, D., Wets, W., Brijs, T., Vanhoof, K., Arentze, T., Timmermans, H., 2006. Integrating Bayesian networks and decision trees in a sequential rule-based transportation model. *Eur. J. Oper. Res.*, **175**(1):16-34. [doi:10.1016/j.ejor.2005.03.022]

Li, B., 2009. Markov models for Bayesian analysis about transit route origin-destination matrices. *Transp. Res. Part B*, **43**(3):301-310. [doi:10.1016/j.trb.2008.07.001]

Nassir, N., Khani, A., Lee, S.G., Noh, H., Hickman, M., 2011. Transit stop-level origin-destination estimation through use of transit schedule and automated data collection system. *Transp. Res. Rec.*, **2263**:140-150. [doi:10.3141/2263-16]

Pelletier, M.P., Trépanier, M., Morency, C., 2011. Smart card data use in public transit. *Transp. Res. Part C*, **19**(4):557-568. [doi:10.1016/j.trc.2010.12.003]

Rahbee, A.B., 2009. Farecard passenger flow model at Chicago transit authority, Illinois. *Transp. Res. Rec.*, **2072**: 3-9. [doi:10.3141/2072-01]

Reddy, A., Lu, A., Kumar, S., Bashmakov, V., Rudenko, S., 2009. Entry-only automated fare collection (AFC) system data used to infer ridership, rider destinations, unlinked trips, and passenger miles. *Transp. Res. Rec.*, **2110**:128-136. [doi:10.3141/2110-16]

Trépanier, M., Tranchant, N., Chapleau, R., 2007. Individual trip destination estimation in a transit smart card automated fare collection system. *J. Intell. Transp. Syst.*, **11**(1):1-14. [doi:10.1080/15472450601122256]

Trépanier, M., Morency, C., Agard, B., 2009. Calculation of transit performance measures using smartcard data. *J. Publ. Transp.*, **12**(1):79-96.

US Energy Information Administration, 2007. International Energy Outlook 2007. Available from http://www.eia.gov/forecasts/archive/ieo07/index.html [Accessed on Feb. 23, 2010].

Zhang, L., Zhao, S., Zhu, Y., Zhu, Z., 2007. Study on the Method of Constructing Bus Stops OD Matrix Based on IC Card Data. Int. Conf. on Wireless Communications, Networking and Mobile Computing, p.3147-3150. [doi:10.1109/WICOM.2007.780]

Zhang, Y.F., 2002. Programming on OD Matrix Estimation—Application in New York City Mass Transit System. Proc. 3rd Int. Conf. on Traffic and Transportation Studies, p.786-792. [doi:10.1061/40630(255)110]

Zhao, J., Rahbee, A., Wilson, N.H.M., 2007. Estimating a rail passenger trip origin-destination matrix using automatic data collection systems. *Comput.-Aided Civ. Infr. Eng.*, **22**(5):376-387. [doi:10.1111/j.1467-8667.2007.00494.x]