# Transit Smart Card Data Mining for Passenger Origin Information Extraction

Xiaolei Ma
Graduate Research Assistant
Box 352700
Department of Civil and Environmental Engineering, University of Washington
Seattle, WA 98195-2700
Tel: (518) 618-6775
Email: xiaolm@uw.edu

Yinhai Wang, Ph.D. (Corresponding author)
Professor
Box 352700
Department of Civil and Environmental Engineering, University of Washington
Seattle, WA 98195-2700
Tel: (206) 616-2696
Fax: (206) 543-1543
Email: yinhai@uw.edu

Feng Chen
Beijing Transportation Research Center
Beijing, 100073
Tel: 86-10-57079824
Fax: 86-10-57079800
Email: chenf@bjtrc.org.cn

Jianfeng Liu
Beijing Transportation Research Center
Beijing, 100073trb
Tel: 86-10-57079836
Fax: 86-10-57079800
Email: ljf@bjtrc.org.cn

## ABSTRACT

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 especially true for Beijing's 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 Markov chain, the algorithm is further optimized and simplified to reduce its computational complexity to linear. 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:** Automated fare collection system, passenger origin inference, Bayesian decision tree, Markov chain

## INTRODUCTION

The United States energy information administration states that more than 50% of commuters drive their own cars to work (*1*). 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 (*2*). This implies that traffic congestion in metropolitan areas can be mitigated if public transit services can 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 (*2*). Collecting such OD data, however, is extremely difficult and expensive using traditional paper-survey-based approaches (*3, 4*). Automatic fare collection (AFC) systems contain rich spatial and temporal information through contactless smart cards with unique ID, which significantly reduce manpower to collect transit passenger OD data. However, most AFC systems are not designed for OD data collection (*5*), hence further data processing and analysis are necessary for passenger information extraction (*6*). The paper presents a Bayesian decision tree based statistical approach to infer the passenger origin from the imperfect SC transaction data, which is the first step for 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, the paper briefly discusses other 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 proposed to infer passenger origin data based on the existing smart data characteristics. This algorithm is then verified in the next section using GPS (Global Positioning System) data with a detailed result analysis. Conclusions of the study are made at the end of this paper.

## PROBLEM STATEMENT

In May 10, 2006, Beijing Transit Incorporated (BTI) (China) started to issue the Beijing Transportation Smart Card to transit riders. If a user pays the transit fare with the Smart Card, up to 60% discount can be received. Such a big discount quickly stimulated the use of smart card. In 2010, more than 90% of the transit users paid their transit trips with their smart cards (*7*). There are a total of 16 million Smart Card (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 potential to form the complete OD matrix of Beijing transit riders. All SC transactions are archived in a database at Beijing Transportation Research Center (BTRC). The high market penetration rate and tremendous daily transactions ensure a great data source, and presents challenges for transit rider OD extraction as well.

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 the SC scan system design (*8*). Compared to the distance-based fare bus riders, flat-rate bus users do not have check-out records. This creates the first challenge in OD extraction: where does a passenger get off a flat-rate bus? Furthermore, the scan system does not save the location and direction information on the 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, such as smart card ID, route number, driver ID, transaction time, remaining balance, transaction amount, etc, it is not impossible 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 (*9*).

Beijing AFC system should not be treated as a special case, and most cities in China also employ the similar AFC system where passengers' origin information is absent, such as Chongqing City (*10*), Nanning City (*11*), Kunming City (*12*). Even in some developing countries, such as Brazil, AFC system doesn't record any boarding location information (*13*). Therefore, a solution for passenger boarding and alighting information extraction is beneficial to those transit agencies with imperfect SC data internationally. 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.

## LITERATURE REVIEW

Many OD matrix inference approaches have been investigated over the past years. Researches on Metropolitan Transit Authority (MTA)'s MetroCard system in New York City (*14, 6*) revealed the feasibility of station-to-station OD matrix generation in the entry-only automatic fare collection subway system. Zhao *et al*. (*8*) and Rahbee (*15*) proposed a transit OD matrix estimation algorithm for origin-only AFC data from Chicago Transit Authority rail system. However, their algorithms primarily focused on the rail system, where boarding at fixed stations are easier to locate than bus transit systems. Pelletier *et al*. (Pelletier *et al*., 2010) undertook a thorough literature review on transit smart card data usage, and they concluded that properly processing SC data can enhance the strategic, tactical, and operational performances for transit agencies Trépanier *et al*. (*16, 17*) conducted several studies on 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 of a transit performance measure 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 (*13*) outlined a process to construct an automated transit OD matrix based on smart card and GPS data

in Brazil. Nassir *et al*. (*18*) integrated APC data, GPS data, transit schedule data with AFC data to estimate the stop-level passenger origin and destination. Similar with Beijing AFC system, City of Changchun's AFC system lacks both boarding and alighting stops, hence Zhang *el al*. designed an on-bus questionnaire to match each passenger's boarding time for origin inference (*19*). 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 existing literature does not identify any approach suitable for passenger OD information extraction from Beijing's SC transaction data. Hence, an algorithm applicable for Beijing's AFC system is highly desired.

## METHDOLOGY

As mentioned in the Problem Statement section, the boarding stop and bus direction information are missing in 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 to 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 because the SC system for the flat-rate bus does not record 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 are able to extract boarding volume changes with time and passengers who made transfers. By mining these data and combining transit route maps, we may be able to accomplish our goal. Therefore, a two-step approach is designed for passenger origin data extraction: smart card data clustering and transit stop recognition. Details of each step are described below.

### Smart Card Data Clustering

#### *Transaction Data Classification*

First of all, 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 interval between their transactions. Thus, the clustered SC transactions can be represented by a time series of check-in passenger volumes at stops as shown in TABLE 1.

**TABLE 1 Examples of Clustered SC transactions**

| Transaction Cluster No. | Stop ID | Stop Name | Total Transactions | Transaction Timestamp | Time Difference |
|---|---|---|---|---|---|
| 1 | Unknown | Unknown | 18 | 5:26:36 | 0:14:26 |
| 2 | Unknown | Unknown | 9 | 5:41:02 | 0:03:16 |
| 3 | Unknown | Unknown | 11 | 5:44:18 | 0:04:35 |
| 4 | Unknown | Unknown | 27 | 5:48:53 | 0:01:00 |

In TABLE 1, total transactions indicate 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 time at this stop and next stop with boarding passengers. Unlike most entry-only AFC systems in the United States, stop name and ID from each transaction are unknown in Beijing's 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 needed is indeed to map each cluster into the corresponding boarding stop ID.

In summary, the smart card data clustering algorithm contains three steps as follows:

1. All transaction data for each bus are sorted by the transaction timestamp in an ascending order.
2. For two consecutive records, if their transaction time difference is within 60 sec, then, 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 in terminus, and for this bus, one bus trip has completed. Next record will be the beginning for the next bus trip.

The result of the clustering process is 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 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 that makes the pattern discovery process very challenging. Furthermore, unfamiliar passengers or passengers boarding from the check-out doors (this happens for very crowded buses) may take longer than 60 seconds 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.

### *Transaction Cluster Sequence Segmentation*

Beijing has a huge transit network with nearly 1,000 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 (*20*). The key principle used in the transfer stop identification is that we assume the alighting stop in the previous route is spatially the closest to the boarding stop for the next route. This is reasonable because most passengers choose the closest stop for transit transfer (*22*).

Assume a passenger *k* makes a transfer from route *i* to route *j*. If any 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, if the transferring location is unique, we can still use the transfer information to identify the transfer bus stop ID and name. In addition, walk distance between the two stops are 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.

### *Data Mining for Transit Stop Recognition*

If we treat each segmented series of transaction cluster 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 have varying distribution of passengers boarding at any given stop and roadway congestion may cost unpredictable delays. Therefore, the unknown pattern recognition is a very challenging issue. Once the unknown pattern is recognized, the boarding stop for any passenger becomes clear.

Bayesian decision tree algorithm is one of the widely used data mining techniques for pattern recognition (*23*). Each node in the Bayesian decision tree is connected through 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. if we denote the current boarding stop ID at time step *k* as $S_k$, and at time step *k*+1, the next boarding stop ID as $S_{k+1}$, according to Bayesian inference theory (*24*), $S_{k+1}$ 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 $\Pr(S_{k+1} \mid S_1, S_2 ... S_k)$ =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 due to the nature of transit route, the probability of passengers boarding at $S_{k+1}$ at time step $k$+1 is only related to whether the last boarding stop was $S_k$ at time step $k$. That is because if the transaction time and corresponding bus location for SC transaction cluster $k$ is known, the next SC transaction cluster $k$+1 only relies on how fast the bus travels during the time period between SC transaction clusters $k$ and $k$+1. In this case, a SC transaction series can be recognized as a Markov chain process. Markov chain is a stochastic process with the property that the next state only relies on the current state. Therefore, $S_{k+1}$ can be rewritten as:

$$S_{k+1} = \arg\max_j(\Pr(S_{k+1} = j \mid S_1, S_2 ... S_k)) = \arg\max_j(\Pr(S_{k+1} = j \mid S_k = i)) \quad \textit{subject to } i < j \tag{2}$$

The single-step Markov transition probability is defined as $\Pr(S_{k+1} = j \mid S_k = i)$, also denoted as $p_{ij}$, with $i$, $j$ 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 **Π** can be simplified as:

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

where $n$=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.

***Transition Matrix Generation***

To recognize the unknown pattern, it is critical to develop a measure to quantify $p_{ij}$, the possibility of next boarding stop being stop $j$ conditioned on the previous boarding stop being $i$. The higher $p_{ij}$ is, the more likely the next SC transaction cluster corresponds to boarding passengers at stop $j$. In other words, $p_{ij}$ represents the probability for 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 stop $i$ to stop $j$ 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 for cluster $k$ is $t_k$, and that for cluster $k$+1 is $t_{k+1}$, then, the bus travel time from time step $k$ to time step $k$+1 is $t_{k+1} - t_k$, and the stop distance between stop $j$ and stop $i$ is $D_{ij}$, then, the average bus travel speed $V_{ij}$ can be expressed as:

$$V_{ij} = \frac{D_{ij}}{t_{k+1} - t_k} \tag{4}$$

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 quantifying $p_{ij}$.

In the speed normal distribution, the mean travel speed $\mu_{ij}$ and standard deviation $\sigma_{ij}$ 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 stop $i$ and stop $j$, we can calculate the mean travel speed $\mu_{ij}$ and standard deviation $\sigma_{ij}$ as a priori information. It is noteworthy that the speed mean and standard deviation are not dependent on GPS data, but can be also obtained by other data sources such as distance-based-rate SC transaction data. A sensitivity analysis further demonstrates the algorithm's robustness even with different speed data sources.

Then, the transition probability can be reformulated as:

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

where $Z_{ij} = \frac{V_{ij} - \mu_{ij}}{\sigma_{ij}}$, which is the standardized travel speed between stop $j$ and stop $i$

$\Delta$ is a small increase value for travel speed, and it will not impact the algorithm result, since this is a common term for each transition probability.

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

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

Here, $\overline{P}(k+1) = \sqrt[k+1]{\prod_{n=1}^{k} \Pr(S_{n+1} = j \mid S_n = i)}$ denotes the geometric mean probability of passengers boarding stop sequence at time step $k$+1. It is also the probability for the identified stop sequence to match the unknown pattern.

## ALGORITHM IMPLEMENTATION AND OPTIMIZATION

### Implementation

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

1. Direction identification

   Beijing transit AFC system doesn't 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. Then the probability of the most likely stop sequence from each of trees will be compared and the one with the highest path probability wins.

2. Outlier removal

   As mentioned in the Smart Card Data Clustering section, in some cases, the delayed transactions impact the accuracy of 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, if one passenger boarded the bus and paid the fare at 8:00 AM, another passenger swiped his smart card to alight at 8:10 AM. Due to the relative 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 there exists a probability that a passenger may retain in the same stop. If the previous stop ID is defined as $i$, the number of total stops in each possible direction is denoted as $N$, and the probability that a passenger stay at stop $i$ in the next time step can be expressed as:

$$p_{ii} = 1 - \sum_{j=i+1}^{j \le N} p_{ij} \tag{8}$$

The probability is able to better depict the situation where passengers may delay a certain period to swipe their Smart Cards during boarding.

3. Bus trip detection

The journey begins 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 Beijing's 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 Bayesian decision tree may be misidentified if the bus trip is mistakenly detected. The solution to this issue is to model the travel time probability of each transaction cluster series. As indicated in the transaction cluster sequence segmentation section, each series is bounded by possible inferred stops, by calculating the travel time for multiple combinations of inferred stops, and comparing with the actual time difference, we are able to determine the most likely bus trip based on the highest probability.

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

## Computational Performance Optimization

Although we illustrated the mathematical form for Markov chain based Bayesian decision tree in theory, this algorithm presented above has not been applied in the real dataset. Cooper (*25*) has proven Bayesian decision tree algorithm a NP (Non-deterministic Polynomial)-hard problem, which means this algorithm cannot be solved in a polynomial time. Conventional approach to calculate 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 as follows:

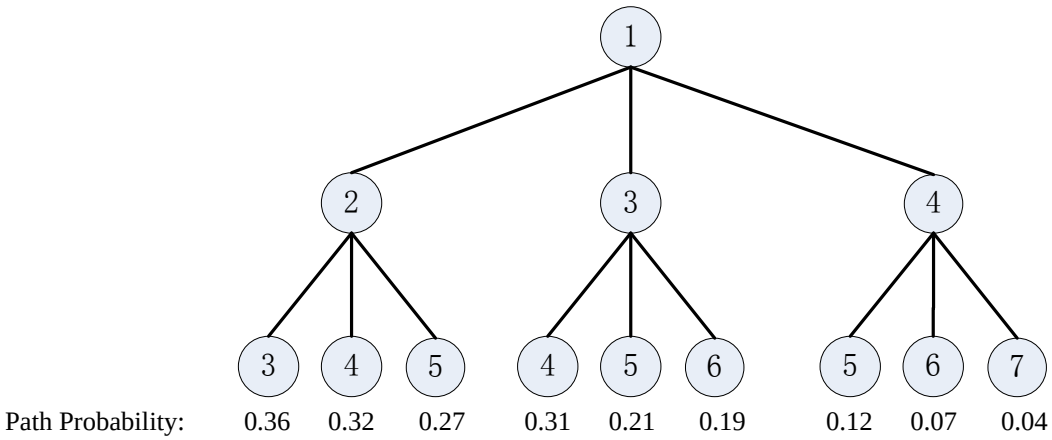


**FIGURE 1 A Bayesian Decision Tree Algorithm Example**

Assume the initial boarding stop is 1. The potential stops in the next step could be stop 2, stop 3, or stop 4 because 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 the time step 3, there are two or more paths ending with stop 3, 4 and 5. Before carrying on the computation in the next time step, we can compare the probability of the paths with the same ending stop, and choose the maximum one, which is also called the partial best path, that is:

In the 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 the Markov Chain model states that the probability of current state given a previous state sequence depends only on the previous state. Hence, five paths calculated in time step 3 guarantees the most probable paths in time step 4 without extra computations of other paths. According to Equation (6), we can express the optimized procedure in mathematics as:

$$\overline{P}(k+1) = \max_{i,j}(\overline{P}(k)(\sqrt[k+1]{\Pr(S_{k+1} = j \mid S_k = i)})) \tag{9}$$

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 computational expensive than calculating the probabilities for all sequences. If we denoted the total stops for a specific route as *n*, and the SC transactions are classified in *m* clusters, which correspond to $m$ time steps in Bayesian decision trees, then the computational complexity for the exhaustive approach can be written as $O(m^n)$. While using the optimized algorithm, the computational complexity is only $O(mn)$. With the optimization, the algorithm can be solved in a finite time, and can be efficiently applied in reality.

## VALIDATION AND ANALYSIS

By installing GPS receivers on flat-rate buses, we can collect the geospatial information and spot speed data in a real-time manner. There are approximately 30% buses equipped with GPS devices in Beijing, and GPS data are updated every 30 seconds. 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 feature 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 the ‘ground truth’ data for the comparison purpose.

In this section, the Markov chain based Bayesian decision tree algorithm is first validated using GPS data for route 22, and then, a sensitivity analysis is conducted to investigate impacts of speed mean and standard deviation calculated from GPS data and other data sources.

### Algorithm validation

Flat-rate based route 22 was selected to infer unknown boarding location, and GPS data associated with route 22 was also collected to verify the result. The SC transaction data and GPS data are all recorded on April 7, 2010. Route 22 contains total 34 stops both inbound and outbound directions as shown in FIGURE 2.

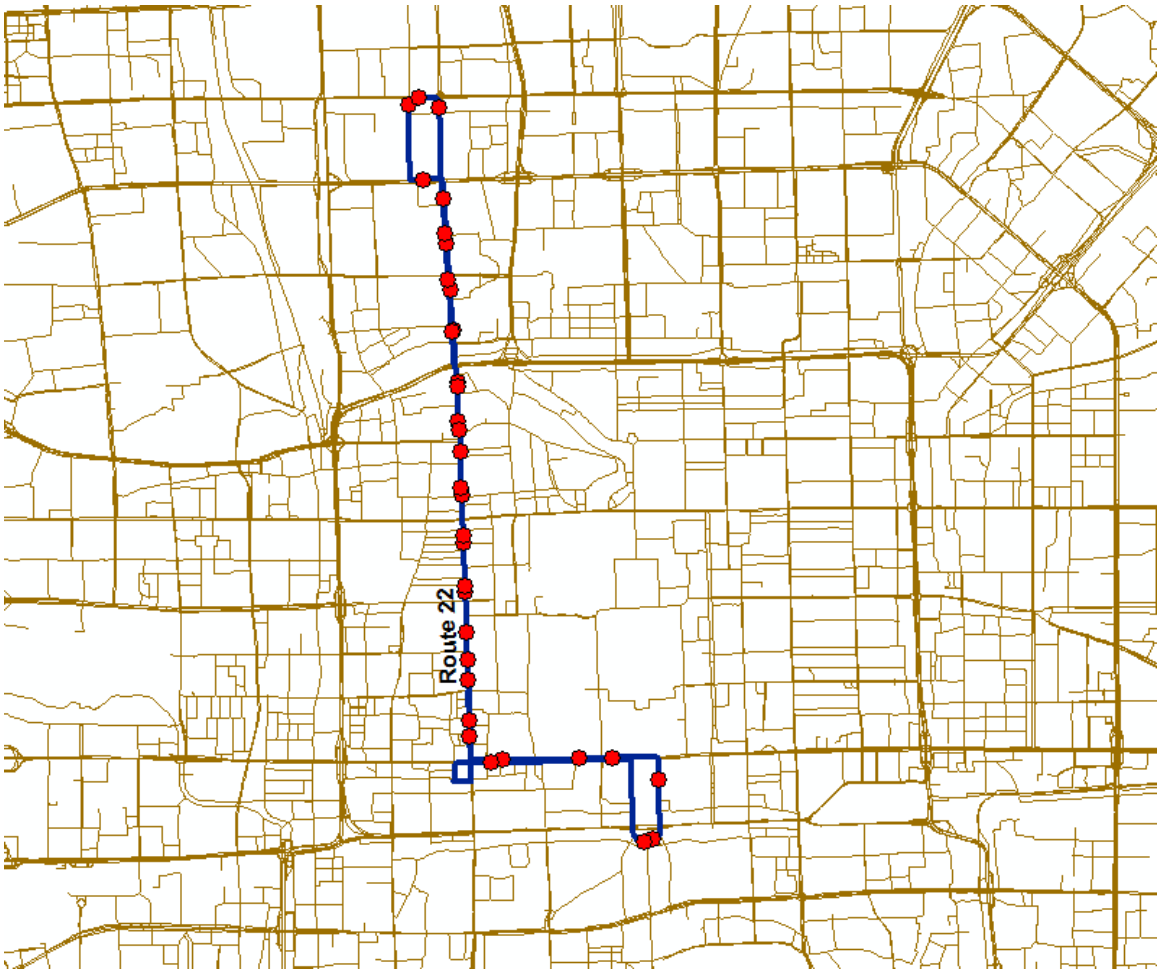


**FIGURE 2 Route 22 in Beijing transit network**

The algorithm results are listed as in TABLE 2. In TABLE 2, there are a total of 12,675 SC transactions mapped with GPS data for Route 22. 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. For Route 22, 95% passenger boarding stops were deducted by the algorithm. 58.6% of results perfectly match with the stops inferred by GPS accurately. There are 11,645 recognized boarding stops within three stops in error, accounting for approximately 96.7% of the total identified stops or 91.6% of all the stops need.

The results are very encouraging. In Beijing's transit network, the error within three stops is acceptable for transit planning level study, since these stops are mostly affiliated with the same traffic analysis zone (TAZ).

**TABLE 2 Results of Bayesian Decision Tree Algorithm for Route 22 Based on GPS Speed**

| Route 22 | Number of Records | Accumulated Percentage in inferred stops | Accumulated Percentage in total stops |
|---|---|---|---|
| Stop ID errors<1 | 7062 | 58.6% | 55.8% |
| Stop ID errors<2 | 10371 | 86.1% | 81.8% |
| Stop ID errors<3 | 11341 | 94.2% | 89.5% |
| Stop ID errors<4 | 11645 | 96.7% | 91.9% |
| Total | 12043 | N/A | 95% |

## Travel speed sensitivity analysis

Recall that in computing the transition matrix, mean travel speed $\mu$ and standard deviation $\sigma$ were extracted from GPS data. However, there are still many flat-rate routes without GPS devices. To understand how the algorithm result changes when the travel speed mean and standard deviation are inaccurate, a sensitivity analysis is carried out for this purpose. Below table show the results when the mean travel speed and standard are retrieved from the distance-based fare routes, and these routes share common stops with the "no-GPS" flat-fare route. Since boarding stop and alighting stop are known in the distance-based fare buses, using the distance between two known stop and time difference, we are able to extract the mean and variance of travel speed to construct the transition matrix.

**TABLE 3 Results of Bayesian Decision Tree Algorithm for Route 22 Based on the Speed from Distance-based Fare Routes**

| Route 22 | Number of Records | Accumulated Percentage in inferred stops | Accumulated Percentage in total stops |
|---|---|---|---|
| Stop ID errors<1 | 6841 | 58.5% | 54% |
| Stop ID errors<2 | 10319 | 88.2% | 81.4% |
| Stop ID errors<3 | 11296 | 96.6% | 89.1% |
| Stop ID errors<4 | 11509 | 98.4% | 90.8% |
| Total | 11694 | N/A | 92.2% |

Different data sources only slightly influence the percentage of inferred stops. 92.2%

boarding stops can be extracted using the speed generated from distance-based fare routes, but the accuracy within three stop errors is higher than the one using GPS speed. The result indicates the algorithm is not sensitive to the travel speed, even without GPS data, we are still able to correctly identify passenger boarding stops using other data sources. This is not surprising, because in normal distribution, mean and standard only influence the shape for probability density function, as long as we make a reasonable assumption for bus travel speed calculation, the algorithm results will not fluctuate significantly.

## CONCLUSIONS

Different from most entry-only AFC systems in other countries, Beijing's AFC system does not record boarding location information when passengers get on the buses and swipe their smart cards. 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 travel speed normal distribution probability density function is used to measure the randomness of passenger boarding stops, where its mean and variance are not sensitive to the algorithm accuracy and thereby not dependent on other data sources. Moreover, we can use the time invariance of Markov chain model to further reduce the computational complexity of the algorithm to linear load. The optimized algorithm is proven effective using the SC transaction data from two routes.

However, this algorithm can still be improved in many ways; for instance, the algorithm does not perform well under the circumstance that 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 functions, e.g., the closeness between each stop and 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 great start for mining transit passenger ODs from the SC transaction data for transit system planning and operations.

## ACKNOWLEDGEMENTS

The authors would like to appreciate Beijing Science and Technology Committee for funding support. All data used for this study were provided by Beijing Transportation Research Center (BTRC). We are grateful to BTRC for their data supports.

## REFERENCE

1. US Energy Information Administration, International Energy Outlook 2007. Accessed online at http://www.eia.gov/forecasts/archive/ieo07/index.html, on Nov. 2, 2010.
2. Li, B., Markov models for Bayesian analysis about transit route origin-destination matrices. *Transportation Research Part B*, 2009, Vol. 43, No. 3, pp. 301-310.

3. Reddy, A., Lu, A., Kumar, S., Bashmakov, V., Rudenko, S., “Application of Entry-Only Automated Fare Collection (AFC) System Data to Infer Ridership, Rider Destinations, Unlinked Trips, and Passenger Miles”, Preprint CD-ROM for the 88th Annual Meeting of Transportation Research Board, Washington, D.C. 2009.
4. Hofmann, M., Wilson, S., White, P., “Automated Identification of Linked Trips at Trip Level Using Electronic Fare Collection Data”, Preprint CD-ROM for the 88th Annual Meeting of Transportation Research Board, Washington, D.C. 2009.
5. Pelletier M-P., Trépanier M., Morency C., “Smart card data use in public transit”, *Transportation Research Part C*, 2011, Vol. 19, Issue 4, pp. 557-568.
6. Barry, J.J., Freimer, R., and Slavin, H. “Use of Entry-Only Automatic Fare Collection Data to Estimate Linked Transit Trips in New York City”, *Transportation Research Record: Journal of the Transportation Research Board*, No. 2112, Transportation Research Board of the National Academies, Washington, D.C., 2009, pp. 53-61.
7. Beijing Transportation Research Center. Beijing Transportation Smart Card Usage Survey. Research Report. 2010.
8. Zhao, J., Rahbee, A. and Wilson, N.H.M. Estimating a rail passenger trip origin-destination matrix using automatic data collection systems. *Computer-Aided Civil And Infrastructure Systems*, 2007, 22, 5,376-387.
9. Zhang Yu-Fang. Programming on Origin-Destination (OD) Matrix Estimation-Application in New York City Mass Transit System, *Proceedings of the Third International Conference on Traffic and Transportation Studies*, 2002, pp. 786-792.
10. Gao, L.X. and Wu, J. P., 2011. An algorithm for Ming Passenger Flow Information from Smart Card Data, Journal of Beijing University of Posts and Telecommunications, Jun. 2011, vol. 34, No.3, pp. 94-97.
11. Chen, J., 2009. Research on Travel Demand Analysis of Urban Public Transportation Based on Smart Card Data Information. Ph.D. dissertation, Tongji University.
12. Zhou, T., Zhai C., and Gao Z., 2007. Approaching Bus OD Matrices based on Data Reduced from Bus IC Cards. May. 2007, vol. 5, No.3, pp. 48-52.
13. Farzin, J. M., “Constructing an Automated Bus Origin-Destination Matrix Using Farecard and Global Positioning System Data in Sao Paulo, Brazil”, *Transportation Research Record: Journal of the Transportation Research Board*, No. 2072, Transportation Research Board of the National Academies, Washington, D.C., 2008, pp. 30-37.
14. Barry, J.J., Newhouser, R., Rahbee, A., and Sayeda, S. “Origin and Destination Estimation in New York City with Automated Fare System Data”, *Transportation Research Record: Journal of the Transportation Research Board*, No. 1817, Transportation Research Board of the National Academies, Washington, D.C., 2002, pp. 183-187.
15. Rahbee, A.B. “Farecard Passenger Flow Model at Chicago Transit Authority, Illinois”, *Transportation Research Record: Journal of the Transportation Research Board*, No. 2072, Transportation Research Board of the National Academies, Washington, D.C., 2009, pp. 3-9.

16. Trépanier, M., Tranchant, N., Chapleau, R., “Individual Trip Destination Estimation in a Transit Smart Card Automated Fare Collection System”, *Journal of Intelligent Transportation Systems*, 11(1):1–14, 2007.
17. Trépanier, M., Morency, C., Agard, B., “Calculation of Transit Performance Measures Using Smartcard Data”, *Journal of Public Transportation*, 2009, Vol. 12, No. 1.
18. Nassir, N., Khani A., Lee, S. G., Noh, H., and Hickman, M., Transit Stop-level O-D Estimation Using Transit Schedule and Automated Data Collection System, Preprint CD-ROM for the 90th Annual Meeting of Transportation Research Board, Washington, D.C. 2011.
19. Zhang, L., Zhao, S., Zhu, Y., and Zhu, Z., Study on the Method of Constructing Bus Stops OD Matrix Based on IC Card Data. Wireless Communications, Networking and Mobile Computing WiCom 2007, pp. 3147-3150
20. Jang, W., “Travel Time and Transfer Analysis Using Transit Smart Card Data”, *Transportation Research Record: Journal of the Transportation Research Board*, No. 2144, Transportation Research Board of the National Academies, Washington, D.C., 2010, pp. 142-129.
21. Seaborn, C., Wilson, N. H. M., Attanucci, J., “Using Smart Card Fare Payment Data To Analyze Multi-Modal Public Transport Journeys (London, UK)”, Preprint CD-ROM for the 88th Annual Meeting of Transportation Research Board, Washington, D.C. 2009.
22. Chu, K. K. A. and Chapleau, R., “Enriching Archived Smart Card Transaction Data for Transit Demand Modeling”, *Transportation Research Record: Journal of the Transportation Research Board*, No. 2063, Transportation Research Board of the National Academies, Washington, D.C., 2008, pp. 63-72.
23. Janssens, D., Wets, W., Brijs, T., Vanhoof, K., Arentze, T., Timmermans, H., “Integrating Bayesian networks and decision trees in a sequential rule-based transportation model”, *European Journal of Operational Research*, 2006, 175, pp. 16-34.
24. Bayes, Thomas; Price, Mr. "An Essay towards solving a Problem in the Doctrine of Chances.". *Philosophical Transactions of the Royal Society of London* 53 (0): 370–418, 1763.
25. Cooper, G. F., “The computational complexity of probabilistic inference using Bayesian belief networks”, *Artificial Intelligence*, Vol. 42, pp. 393-405, 1990.