In: Data Mining
Editor: Harold L. Capri

ISBN: 978-1-63463-738-1

Chapter 1

TRANSIT PASSENGER ORIGIN INFERENCE USING SMART CARD DATA AND GPS DATA

Xiaolei Ma*[*1], Ph.D. and Yinhai Wang[2], Ph.D.
[1]School of Transportation Science and Engineering,
Beihang University, Beijing, China
[2]Department of Civil and Environmental Engineering,
University of Washington, Seattle, WA, US

ABSTRACT

To improve customer satisfaction and reduce operation costs, transit authorities have been striving to monitor their transit service quality and identify the key factors to attract the transit riders. Traditional manual data collection methods are unable to satisfy the transit system optimization and performance measurement requirement due to their expensive and labor-intensive nature. The recent advent of passive data collection techniques (e.g., Automated Fare Collection and Automated Vehicle Location) has shifted a data-poor environment to a data-rich environment, and offered the opportunities for transit agencies to conduct comprehensive transit system performance measures. Although it is possible to collect highly valuable information from ubiquitous transit data, data usability and accessibility are still difficult. Most Automatic Fare Collection (AFC) systems are not designed for transit performance monitoring, and additional passenger trip information cannot be directly

* Email: xiaolm@uw.edu

retrieved. Interoperating and mining heterogeneous datasets would enhance both the depth and breadth of transit-related studies. This study proposed a series of data mining algorithms to extract individual transit rider's origin using transit smart card and GPS data. The primary data source of this study comes from the AFC system in Beijing, where a passenger's boarding stop (origin) and alighting stop (destination) on a flat-rate bus are not recorded on the check-in and check-out scan. The bus arrival time at each stop can be inferred from GPS data, and individual passenger's boarding stop is then estimated by fusing the identified bus arrival time with smart card data. In addition, a Markov chain based Bayesian decision tree algorithm is proposed to mine the passengers' origin information when GPS data are absent. Both passenger origin mining algorithms are validated based on either on-board transit survey data or personal GPS logger data. The results demonstrates the effectiveness and efficiency of the proposed algorithms on extracting passenger origin information. The estimated passenger origin data are highly valuable for transit system planning and route optimization.

Keywords: Automated fare collection system, transit GPS, passenger origin inference, Bayesian decision tree, Markov chain

INTRODUCTION

According to the Census of 2000 in the United States, approximately 76% people chose privately owned vehicles to commute to work in 2000 (ICF consulting, 2003). Recent studies conducted by the 2009 American Community Survey indicate 79.5% of home-based workers drive alone for commuting (McKenzie and Rapino, 2009). Many developing countries, e.g., China, also rely on privately owned vehicles to commute. For example, more than 34% of the Beijing residents chose cars as their primary travel mode while only 28.2% chose transit in 2010 (Beijing Transportation Research Center, 2012). Public transit has been considered as an effective countermeasure to reduce congestion, air pollution, and energy consumption (Federal Highway Administration, 2002). According to 2005 urban mobility report conducted by Texas Transportation Institute (2005), travel delay in 2003 would increase by 27 percent without public transit, especially in those most congested metropolitan cites of U.S., public transit services have saved more than 1.1 billion hours of travel time. Moreover, public transit can help enhance business, reduce city sprawl through the transit oriented development (TDO). During certain emergency scenarios, public transit can even act as a

safe and efficient transportation mode for evacuation (Federal Highway Administration, 2002). Based on the aforementioned reasons, it is of critical importance to improve the efficiency of public transit system, and promote more roadway users to utilize public transit. To fulfill these objectives, transit agencies need to understand the areas where improvements can be further made, and whether community goals are being met, etc. A well-developed performance measure system will facilitate decision making for transit agencies. Transit agencies can evaluate the transit ridership trends with fare policy changes and identify where and when better transit service should be provided. In addition, transit agencies are also required to summarize transit performance statistics for reporting to either the National Transit Database (Kittelson & Associates et al., 2003), or the general public who are interested knowing how well transit service is being provided. Nevertheless, developing a set of structured performance measures often requires a large amount of data and the corresponding domain knowledge to process and analyze these data. These obstacles create challenges for transit agencies to spend time and effort undertaking. Traditionally, transit agencies heavily rely on manual data collection methods to gather transit operation and planning data (Ma et al., 2012). However, traditional data collection methods (e.g., travel diary, survey, etc.) are fairly costly and difficult to implement at a multiday level due to their low response rate and accuracy. Transit agencies have spent tremendous manpower and resource undertaking manual data collections, and consumed a significant amount of energy and time to post-process the raw data. With advances in information technologies in intelligent transportation systems (ITS), the availability of public transit data has been increasing in the past decades, which has gradually shifted public transit system into a data-rich paradigm. Automatic Fare Collection (AFC) system and Automatic Vehicle Track (AVL) system are two common passive data collection methods. AFC system, also known as Smart Card system, records and processes the fare related information using either contactless or contact card to complete the financial transaction (Chu, 2010). There exist two typical types of AFC systems: entry-only AFC system and distance-based AFC system. In the entry-only AFC system, passengers are only required to swipe their smart cards over the card reader during boarding, while passengers need to check in and check out during both their boarding and alighting procedures for the distance-based AFC system. AVL and AFC technologies hold substantial promise for transit performance analysis and management at a relative low cost. However, historically, both AVL and AFC data have not been used to their full potentials. Many AVL and AFC systems do not archive data in a readily

utilized manner (Furth, 2006). AFC system is initially designed to reduce workloads of tedious manual fare collections, not for transit operation and planning purposes, and thereby, certain critical information, such as specific spatial location for each transaction, may not be directly captured. AVL system tracks transit vehicles' geospatial locations by Global Positioning System (GPS) at either a constant or varying time interval. The accuracy of GPS occasionally suffers from signal loss due to tall building obstructions in the urban area (Ma et al., 2011). Both of the AFC system and AVL system have their inherent drawbacks in monitoring transit system performance, and require analytical approaches to eliminate the erroneous data, remedy the missing values, and mine the unseen and indirect information.

The remainder of this paper is organized as follows: transit smart card data and GPS data are described in the section 2. Based on these data sets, a data fusion method is initially proposed to integrate with roadway geospatial data to estimate transit vehicles arrival information. And then, a Bayesian decision tree algorithm is presented to estimate each passenger's boarding stop when GPS data are unavailable. Considering the expensive computational burden of decision tree algorithms, Markov-chain property is taken into account to reduce the algorithm complexity. On-board survey and GPS data from the Beijing transit system are used to test and verify the proposed algorithms. Conclusion and future research efforts are summarized at the end of this paper.

RESEARCH BACKGROUND

Data from AFC system and AVL system are the two primary sources in this study. Beijing Transit Incorporated began to issue smart cards in May 10, 2006. The smart card can be used in both the Beijing bus and subway systems. Due to discounted fares (up to 60% off) provided by the smart card, more than 90% of the transit riders pay for their transit trips with their smart cards in 2010 (Beijing Transportation Research Center, 2010). Two types of AFC systems exist in Beijing transit: flat fare and distance-based fare. Transit riders pay at a fixed rate for those flat fare buses when entering by tapping their smart cards on the card reader. Thus, only check-in scans are necessary. For the distance-based AFC system, transit riders need to swipe their smart cards during both check-in and check-out processes. Transit riders need to hold their smart cards near the card reader device to complete transactions when entering or exiting buses. Smart card can be used in Beijing subway system as well, where passengers need to tap their smart card on top of fare gates during

entering and existing subway stations. Both boarding and alighting information (time and location) are recorded by the fare gates. Although transit smart card exhibits its superiority on its convenience and efficiency, there are still the following issues to prevent transit agencies fully taking advantages of smart card for operational purposes:

- Passenger boarding and alighting information missing

Due to a design deficiency in the smart card scan system, the AFC system on flat fare buses does not save any boarding location information, whereas the AFC system stores boarding and alighting location, except for boarding time information on distance-based fare buses. Key information stored in the database includes smart card ID, route number, driver ID, transaction time, remaining balance, transaction amount, boarding stop (only available for distance-based fare buses), and alighting stop (only available for distance-based fare buses).

- Massive data sets

More than 16 million smart card transactions data are generated per day. Among these transactions, 52% are from flat-rate bus riders. These smart card transactions are scattered in a large-scale transit network with 52386 links and 43432 nodes as presented in figure 1:

Figure 1. Beijing Transit GIS Network.

- Limited external data with poor quality

Only approximate 50% of transit vehicles in Beijing are equipped with GPS devices for tracking. GPS data are periodically sent to the central server at a pre-determined interval of 30 seconds. However, the collected GPS data suffer from two major data quality issues: (1) vehicle direction information is missing; (2) GPS points fluctuation (Lou, et al., 2009). Map matching algorithms are needed to align the inaccurate GPS spatial records onto the road network. In addition, most of transit routes are not designed to have fixed schedules because of high ridership demands, and only certain routes with a long distance or headway follow schedules at each stop (Chen, 2009). The above characteristics of the Beijing AFC and AVL systems create more challenges to process and mine useful information.

It is noteworthy that the AFC system used in Beijing is not a unique case. Most cities in China also employ the similar AFC system where passengers' origin information is absent, such as Chongqing City (Gao and Wu, 2011), Nanning City (Chen, 2009), Kunming City (Zhou et al., 2007). In other developing countries, such as Brazil, AFC system does not record any boarding location information as well (Farzin, 2008). Therefore, a solution for passenger boarding and alighting information extraction is beneficial to those transit agencies with imperfect SC data internationally.

TRANSIT PASSENGER ORIGIN INFERENCE

Because smart card readers in the flat-rate buses do not record passengers' boarding stops, it is desired to infer individual boarding location using smart card transaction data. In this section, two primary approaches are presented to achieve this goal. Approximately 50% transit vehicles are equipped with GPS devices in Beijing entry-only AFC system. Therefore, a data fusion method with GPS data, smart card data and GIS data is firstly developed to estimate each bus's arrival time at each stop and infer individual passenger's boarding stop. And then, for those buses without GIS devices, a Bayesian decision tree algorithm is proposed to utilize smart card transaction time and apply Bayesian inference theory to depict the likelihood of each possible boarding stop. In order to expand the usability of proposed Bayesian decision tree algorithm in large-scale datasets, Markov chain optimization is used to reduce the algorithm's computational complexity. Both two transit passenger origin

inference algorithms are validated using external data (e.g., on-board survey data and GPS data).

Passenger Origin Inference with GPS Data

In the first step, a GPS-based arrival information inference algorithm is presented to estimate the arrival time for each transit stop, and then, the inferred stop-level arrival time will be matched with the timestamp recorded in AFC system. The temporally closest smart card transaction record will be assigned with each known stop ID. The logic flow chart is demonstrated in Figure 2. The major data processing procedure will be detailed below.

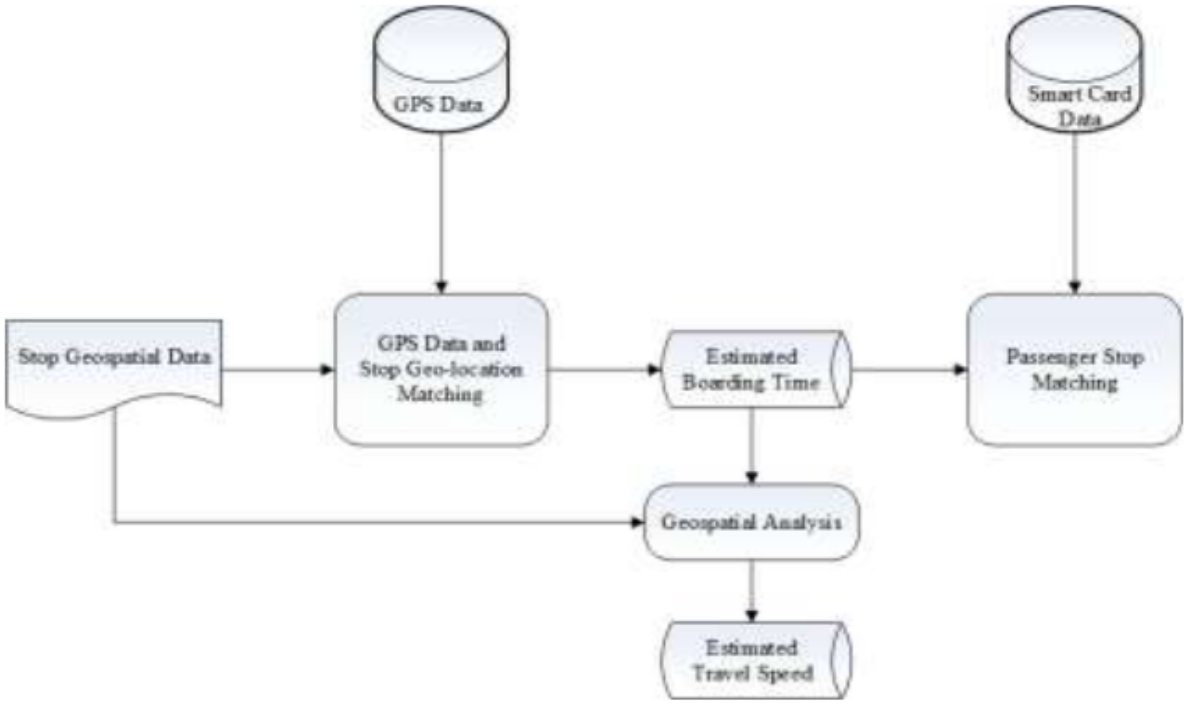


Figure 2. Flow Chart for Passenger Origin Inference with GPS Data.

Bus Arrival Time Extraction

Three primary data sources are involved in the passenger information extraction: vehicle GPS data; transit stop spatial location data; and flat-fare-based smart card transaction data. A transit GIS network contains the geospatial location of each stop for any transit routes. The GPS device mounted in the bus can record each bus's location and timestamp every 30 seconds, but the data quality of collected GPS records is not satisfying: No directional information is recorded in Beijing AVL system; GPS points are off

the roadway network due to the satellite signal fluctuation. Data preprocessing is required prior to bus arrival time estimation. A program is written to parse and import raw GPS data into a database in an automatic manner. Key fields of a GPS record are shown in Table 1.

Table 1. Examples of GPS raw data

Vehicle ID	Date time	Latitude	Longitude	Spot speed	Route ID
00034603	2010-04-07 09:28:57	39.73875	116.1355	9.07	00022
00034603	2010-04-07 09:29:27	39.73710	116.1358	14.26	00022
00034603	2010-04-07 09:29:58	39.73592	116.1357	19.63	00022
00034603	2010-04-07 09:30:28	39.73479	116.1357	0	00022
00034603	2010-04-07 09:30:58	39.73420	116.1357	3.52	00022

The first step is to estimate the bus arrival time for each stop by joining GPS data and the stop-level geo-location data. A buffer area can be created around each particular stop for a certain transit route using the GIS software. Within this area, several GPS records are likely to be captured. However, identifying the geospatially closest GPS record to each particular stop is challenging since there could be a certain number of unknown directional GPS records within the specified buffer zone. Thanks to the powerful geospatial analysis function in GIS, each link (i.e., polyline) where each transit stop is located is composed of both start node and end node, and this implies that the directional information for each GPS record is able to infer by comparing the link direction and the direction changes from two consecutive GPS records. With the identified direction, the distance from each GPS point to this particular stop can be calculated, and the timestamp with the minimum distance will be regarded as the bus arrival time at the particular stop. Figure 2 visually demonstrates the above algorithm procedure. Inbound stop represents the physical location of a particular transit stop, and this stop is snapped to a transit link, whose direction is regulated by both a start node and an end node. By comparing the driving direction from GPS records with the link direction, the nearest GPS records to this particular stop can be identified, and marked by the red five-pointed star on the map. The timestamp associated with this five-pointed star will be considered as the arrival time for this inbound stop. The

merit of the bus arrival time estimation algorithm lies in its efficiency. Rather than searching all the GPS data to identify the traveling direction for each stop, the proposed algorithm shrinks down the searching area, and filters out those unlikely GPS data. The operation greatly alleviates the computational burden, and is relatively easy to implement in the large-scale datasets, which is particularly critical to process the tremendous amount of datasets within an acceptable time period.

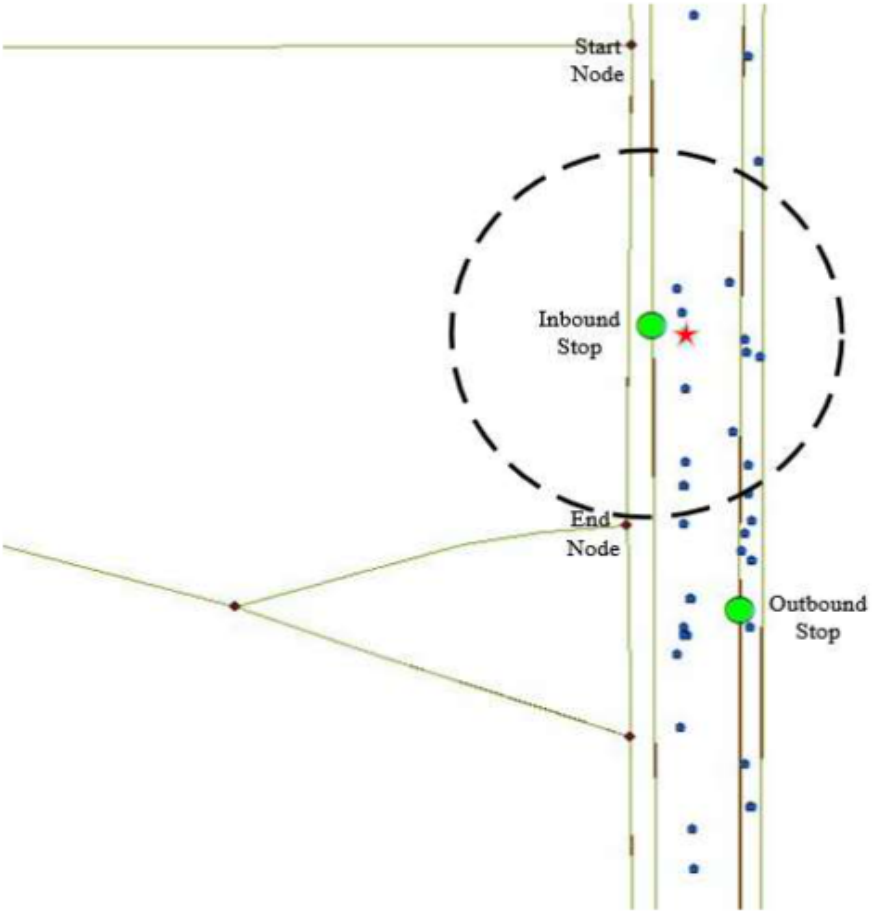
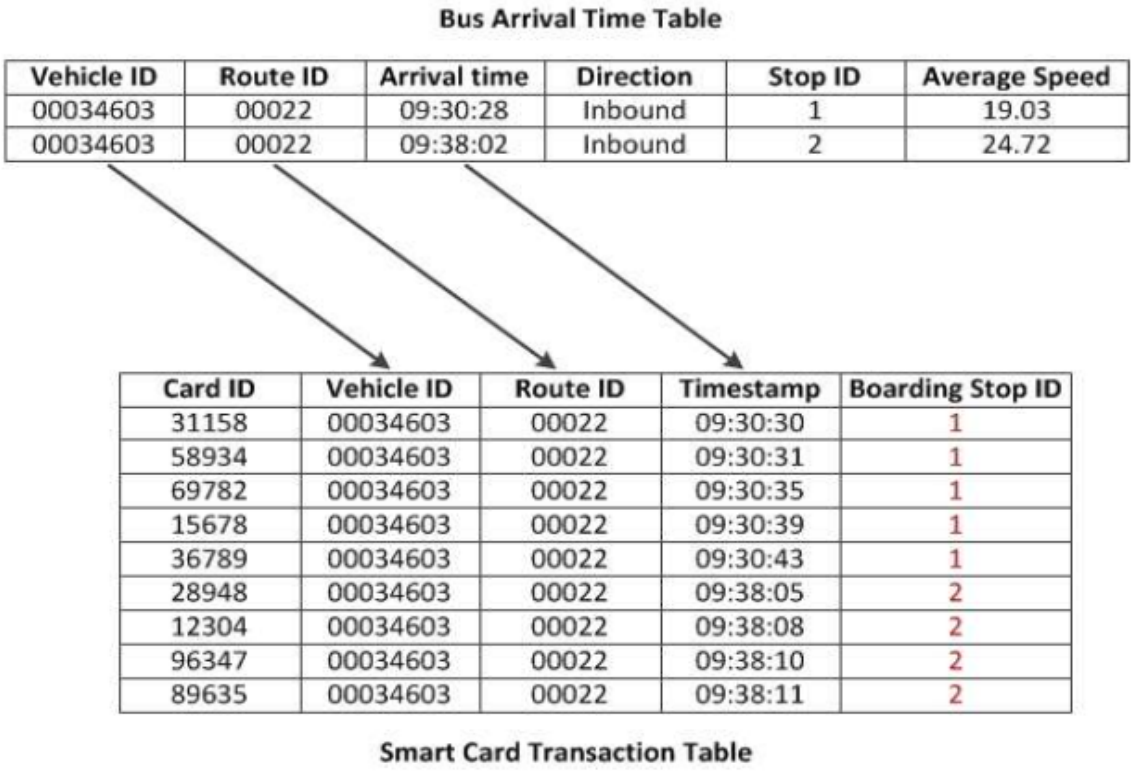


Figure 3. Boarding Time Estimation with GPS Data and Transit Stop Location Data.

Passenger Boarding Location Identification with Smart Card Data

For each smart card data transaction record, the boarding stop can be estimated by matching the recorded timestamp and the identified bus arrival time. As presented in Figure 4, for each smart card transaction record, the transaction time is compared with the inferred bus arrival time at each stop. This record will be assigned to a particular stop where the bus arrival time is the most temporally closed with its transaction time. Since passengers begin to embark the bus at a relative short time interval, this data fusion method is able to capture almost all missing boarding stops.



Bus Arrival Time Table

Vehicle ID	Route ID	Arrival time	Direction	Stop ID	Average Speed
00034603	00022	09:30:28	Inbound	1	19.03
00034603	00022	09:38:02	Inbound	2	24.72

Card ID	Vehicle ID	Route ID	Timestamp	Boarding Stop ID
31158	00034603	00022	09:30:30	1
58934	00034603	00022	09:30:31	1
69782	00034603	00022	09:30:35	1
15678	00034603	00022	09:30:39	1
36789	00034603	00022	09:30:43	1
28948	00034603	00022	09:38:05	2
12304	00034603	00022	09:38:08	2
96347	00034603	00022	09:38:10	2
89635	00034603	00022	09:38:11	2

Smart Card Transaction Table

Figure 4. Boarding Stop Identification with Bus Arrival Time.

In addition, because all the arrival time for all stops of a particular transit route can be estimated, the average travel time between two adjacent stops can be calculated as well. This speed statistics is not only critical for transit performance measures, but also provides prior information for passenger origin inference when GPS data are absent.

Validation

Compared with bus arrival time, door opening time can be more accurately matched with smart card transaction time. This is because each bus may not exactly stop at each transit stop for passenger boarding. The inferred bus arrival time is subject to incur errors when it is used to match with smart card data. To validate the accuracy of the proposed data fusion algorithm for passenger origin inference, on-board transit survey was undertaken to collect bus door opening time and arrival location for each stop of route 651 on January, 13th, 2013. Hand holding GPS devices were used to track the geospatial location of moving buses every 15 seconds. The survey duration was from 8:00 AM to 1: 00 PM, and a total of 75 bus door opening time was manually recorded. These bus door opening time records were then compared with smart card transactions from 417 passengers, and these estimated stops can be considered as the ground-truth data. By comparing the ground-truth

data with the results from the proposed GPS data fusion approach, 406 boarding stops were accurately inferred and 11 boarding stops differ from the ground-truth data within one-stop-error range. The proposed algorithm demonstrates its accuracy as high as 97.4%.

Passenger Origin Inference with Smart Card Data

There are still a fair amount of buses without GPS devices, and thus the bus arrival time at each transit stop is not directly measured. However, most passengers scan their cards immediately when boarding and almost all passengers should complete 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. To implement the proposed algorithm in an efficient manner, a Markov Chain based optimization approach is applied to reduce the computational complexity.

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 2.

Table 2. 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 2, 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:

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

Step 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.

Step 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 (Jang, 2010). Two key principles used in the transfer stop identification are:

(1) We assume the alighting stop in the previous route is spatially and temporally the closest to the boarding stop for the next route. This is reasonable because most passengers choose the closest stop for transit transfer within a short period of time (Chu, 2008). Assume a passenger k makes a transfer from route i to route j within n minutes. If route i is a distance-based-rate bus line or a subway line, then we can identify the transfer station that is also the boarding stop of route j. 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 this study, the transfer time duration n is 30 minutes, and the maximum distance between two transfer stops is 300 meters.

(2) We assume that both the alighting time and the boarding time for each particular stop is similar. In this case, we can substitute a passenger boarding stop with another passenger alighting stop. Assume a passenger k makes a transfer from route i to route j. If route j is a subway line, where both its boarding location and time are available, then we can estimate the passenger k's alighting stop of route i, and this alighting stop can be also considered as the boarding stop for those passengers who get on the bus at the same time.

Walk distance between the two stops should be taken into account for inferring the time when the flat-rate bus arrives at the transfer stop. However, several possible boarding stops may exist due to the unknown direction in the flat-rate smart card transaction, and thus additional data mining techniques are needed to find the boarding stop with the maximum likelihood. These data mining techniques will be detailed in the next section.

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

Bayesian Decision Tree Inference

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 (Janssens et al., 2006). 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 (Bayes and Price, 1763), S_{k+1} can be calculated as:

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

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

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:

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

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:

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

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.

Bayesian Decision Tree Inference

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 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 stop i and stop j, we can calculate the mean travel speed μ_{ij} and standard deviation σ_{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 $Z_{ij} = \dfrac{V_{ij} - \mu_{ij}}{\sigma_{ij}}$, which is the standardized travel speed between stop j and stop i, Δ 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. In practice, to avoid the fast growth of Bayesian decision tree, the transition probability can be bounded by a minimum probability to eliminate those unlikely stops during calculation.

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:

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

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 issues 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:

$$p_{ii} = 1 - \sum_{j=i+1}^{j \le N} p_{ij} \tag{7}$$

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, a transaction cluster sequence can be segmented by several series using aforementioned spatiotemporal transfer relationships. Each identified 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 existence of a bus trip based on the highest probability. Figure 5 demonstrates the procedure of identifying a bus trip.

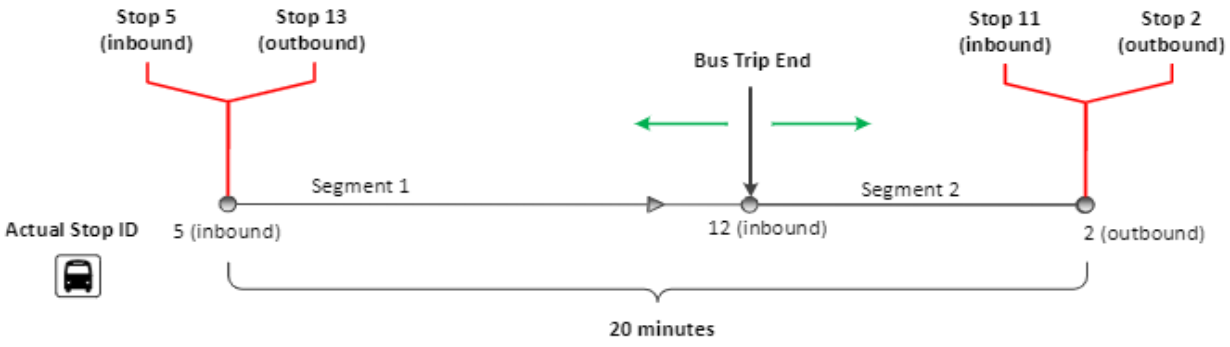


Figure 5. Bus Trip Identification.

As presented in Figure 5, the starting point and ending point of the series can be identified by several possible stops in different directions, and the duration of this transaction cluster series is known as 20 minutes. A variety of trips may exist for this transaction cluster sequence:

Trip 1: The bus travels from the 5th inbound stop to the 11th inbound stop.
Trip 2: The bus travels from the 5th inbound stop to the 2nd outbound stop.
Trip 3: The bus travels from the 13th outbound stop to the 11th inbound stop.
Trip 4: The bus travels from the 13th outbound stop to the 2nd outbound stop.

The maximum and minimum travel time for any trip can be obtained through GPS data or distance-based buses. In addition, the maximum bus layover time can be assumed as 30 minutes. According to the central limit theorem, bus travel time in a known road segment should follow normal distribution, and therefore, we can compute the probability of each scenario, and choose the trip with the maximum probability. If the travel time from stop i to stop j is denoted as t_{ij}, and the probability density function of t_{ij} is defined as:

$$p(t_{ij}) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} \exp\left(-\frac{(t_{ij}-\mu_{ij})^2}{2\sigma_{ij}^2}\right) dt_{ij} \tag{8}$$

where μ_{ij} is the average travel time from stop i to stop j, and σ_{ij} is the standard deviation of travel time from stop i to stop j. If the maximum and minimum travel time (plus maximum and minimum bus layover time) between stop i to stop j are $\max(t_{ij})$ and $\min(t_{ij})$ respectively, then the 95% confidence interval of travel time can be further expressed as:

$$[\mu_{ij} - 1.96\sigma_{ij}, \mu_{ij} + 1.96\sigma_{ij}] = [\min(t_{ij}), \max(t_{ij})] \tag{9}$$

The probability density function of t_{ij} can be rewritten as:

$$p(t_{ij}) = \frac{1}{\sqrt{2\pi\left(\frac{\max(t_{ij})-\min(t_{ij})}{3.92}\right)^2}} \exp\left(-\frac{\left(t_{ij}-\frac{\max(t_{ij})+\min(t_{ij})}{2}\right)^2}{2\left(\frac{\max(t_{ij})-\min(t_{ij})}{3.92}\right)^2}\right) dt_{ij} \tag{10}$$

Each probability for the above four trips can be calculated as 0.54, 0.87, 0.0003 and 0. Therefore, the transaction cluster sequence starts at the 5th inbound stop, and ends at the 2nd outbound stop, and thus a terminus should exist during this trip. This result matched with the actual bus trip. Bayesian decision tree algorithm can be further utilized to infer other uncertain stops within this identified bus trip.

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 (1990) has proven Bayesian decision tree algorithm a NP (Non-deterministic Polynomial)-hard problem, which means that 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:

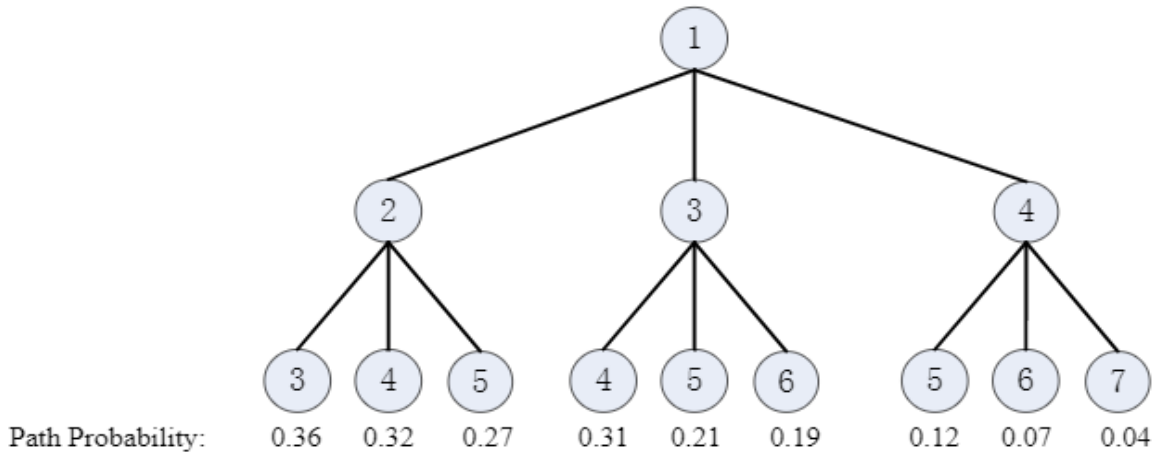


Figure 6. A Bayesian Decision Tree Algorithm Example.

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 that the situations are similar for the remaining stops, a decision tree is fully established. 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.

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 (11), we can express the optimized procedure in mathematics as:

$$\overline{P}(k+1) = \max_{i,j}(\overline{P}(k)(\sqrt[k+1]{\Pr(S_{k+1} = j \mid S_k = i)})) \tag{11}$$

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

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 50% 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. Since the inferred stop ID using GPS data have been validated using the bus on-board survey method, and 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, several sensitivity analyses are conducted to investigate impacts of different parameter settings in Bayesian decision tree. Finally, a computational complexity experiment is also included at the end of this section.

Algorithm Validation

Flat-rate based route 22 was selected to infer unknown boarding location using Markov chain based Bayesian decision tree algorithm, 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. The minimum stop probability is defined as 0.05. If a stop whose transition probability is less than 0.05, then this stop will be abandoned. Route 22 contains a total of 34 inbound and outbound stops as shown in Figure 7.

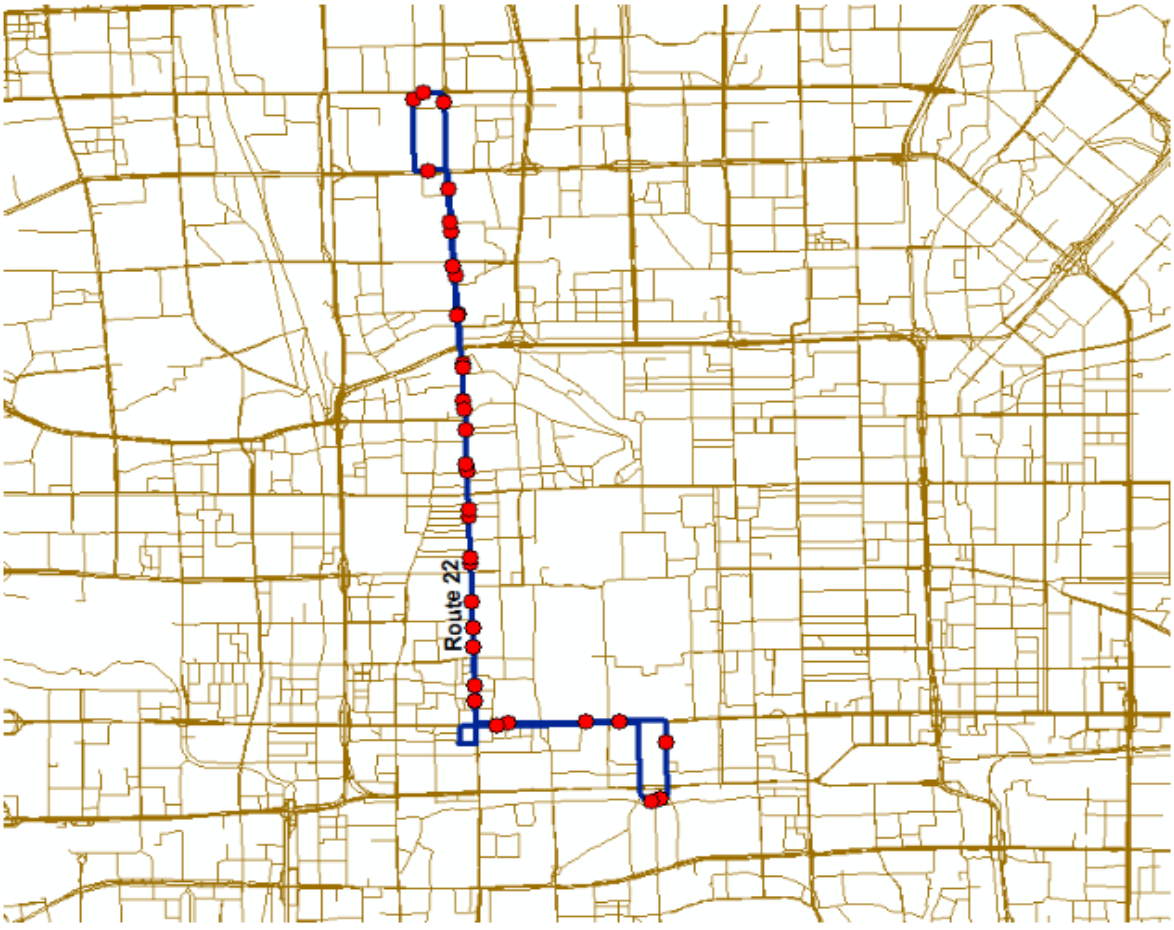


Figure 7. Route 22 in Beijing Transit Network.

The algorithm results are listed as in Table 3 and Figure 8. In Table 3, 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 that are adjacent 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 proposed algorithm. 55.8% of results perfectly match with the stops inferred by GPS accurately. There are 11,645 recognized boarding stops within three-stop distance away from the actual boarding stop, accounting for approximately 96.7% of the total identified records or 91.6% of total records.

Table 3. Results of Bayesian Decision Tree Algorithm for Route 22 Based on GPS Speed

Route 22	Number of records	Accumulated percentage in inferred records	Accumulated percentage in total records
Stop ID error<1	7062	58.6%	55.8%
Stop ID error<2	10371	86.1%	81.8%
Stop ID error<3	11341	94.2%	89.5%
Stop ID error<4	11645	96.7%	91.9%
Total	12043	N/A	97.9%

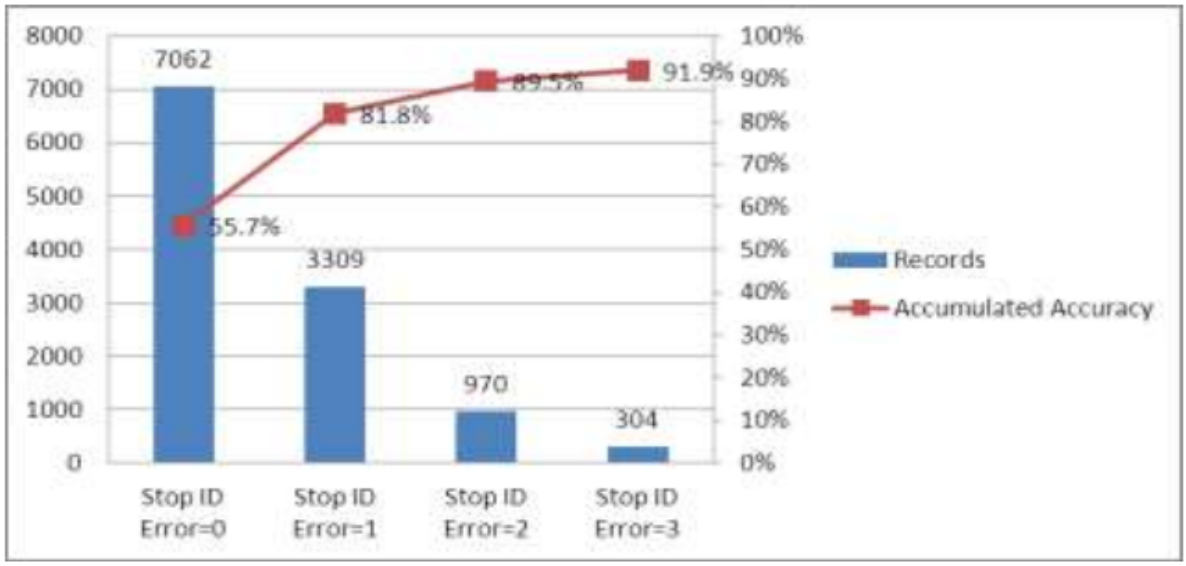


Figure 8. Bayesian Decision Tree Algorithm Accuracy for Route 22 based on GPS Speed.

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) due to the high transit network density.

Sensitivity Analysis

1. Source of travel speed calculation

Recall that in computing the transition matrix, mean travel speed μ and standard deviation σ 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. Table 4 and Figure 9 show the results when the mean and standard deviation of travel speed are retrieved from the distance-based fare routes, and these routes share common stops with the "no-GPS" flat-fare route. Because both boarding stop and alighting stop are known in the distance-based fare buses, we are still able to extract the mean and standard deviation of travel speed between adjacent stops for transition matrix construction.

Table 4. Results of Bayesian Decision Tree Algorithm for Route 22 Based on Speed from Distance-based Fare Routes

Route 22	Number of records	Accumulated percentage in inferred records	Accumulated percentage in total records
Stop ID error<1	6841	58.5%	54%
Stop ID error<2	10319	88.2%	81.4%
Stop ID error<3	11296	96.6%	89.1%
Stop ID error<4	11509	98.4%	90.8%
Total	11694	N/A	92.2%

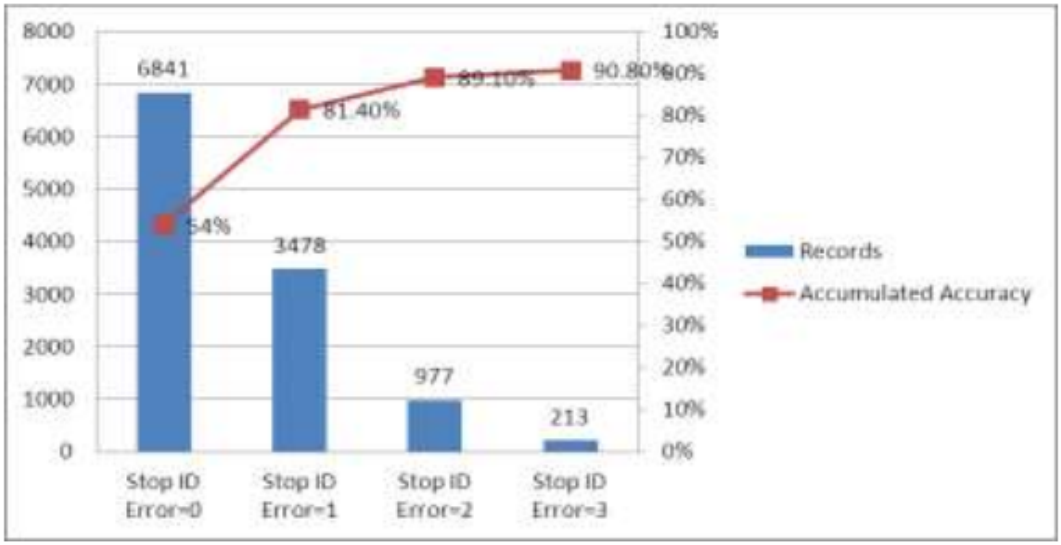


Figure 9. Bayesian Decision Tree Algorithm Accuracy for Route 22 Based on Speed from Distance-based Fare Routes.

Different data sources only slightly influence the percentage of inferred stops. 92.2% boarding stops can be estimated using the speed generated from distance-based fare routes, and the accuracy within three-stop error is 90.8%. The result indicates the proposed 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.

2. Minimum stop probability

Minimum stop probability plays a vital role to impact both the accuracy and efficiency of the proposed algorithm. A too high threshold may eliminate possible boarding stop candidates, and a too low threshold may consume additional computation resources. In this sensitivity analysis, a different minimum stop probability is set as 0.1, which means if the calculated transition probability of a particular stop is lower than 0.1, and then this stop is considered as an unlikely boarding stop. The comparison result is presented in Table 5 and Figure 10.

When the minimum stop probability increases, less boarding stops can be inferred using the proposed algorithm. In addition, the inferred boarding stops are less accurate compared with the ones with minimum stop probability as 0.05. This is a reasonable result since a rigorous probability threshold may limit the prorogation of errors. However, a trade-off exists between algorithm accuracy and efficiency.

Table 5. Results of Bayesian Decision Tree Algorithm for Route 22 with Minimum Stop Probability as 0.1

Route 22	Number of Records	Accumulated Percentage in inferred records	Accumulated Percentage in total records
Stop ID error<1	6011	55.2%	47.4%
Stop ID error<2	9157	84.0%	72.2%
Stop ID error<3	10139	93.1%	80.0%
Stop ID error<4	10589	97.2%	83.5%
Total	10894	N/A	85.9%

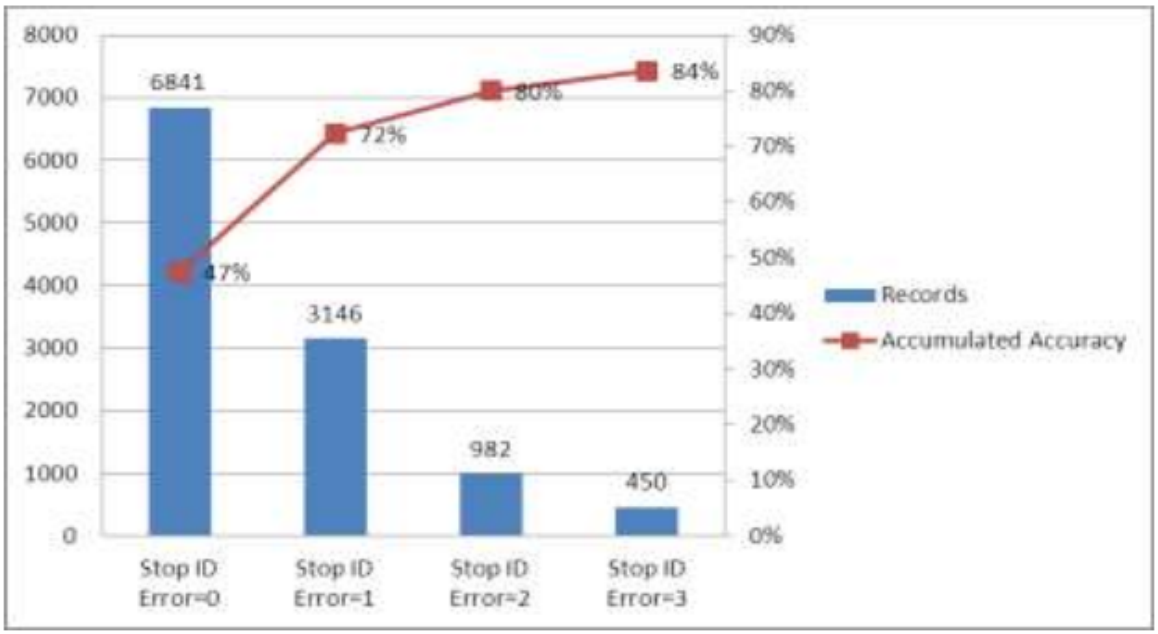


Figure 10. Bayesian Decision Tree Algorithm Accuracy for Route 22 with Minimum Stop Probability as 0.1.

3. Computational complexity comparison

As mentioned in the algorithm optimization section, the computational complexity should be also taken into account when the proposed algorithm is implemented in a large-scale transit network. To compare the algorithm efficiency between the basic Bayesian decision tree algorithm (Basic BDC) and the Markov chain based Bayesian decision tree algorithm (Markov-chain BDC), seven transit routes with an increasing number of total stops are tested. 10,000 smart card transactions for each route on April, 7, 2010 are used for comparison purposes. The experimental result is listed in table 6 and figure 11.

Table 6. Computation Complexity Comparison between Basic and Markov-chain Based Bayesian Decision Tree Algorithms

Route ID	Number of stops	Running time for Basic BDC (milliseconds)	Running time for Markov-chain BDC(milliseconds)
00616	23	3798	493740
00647	36	4890	674820
00005	53	7747	937387
00839	66	17082	1947348
00355	74	21071	2486378
00646	80	23979	4556010
00603	86	29114	5560774

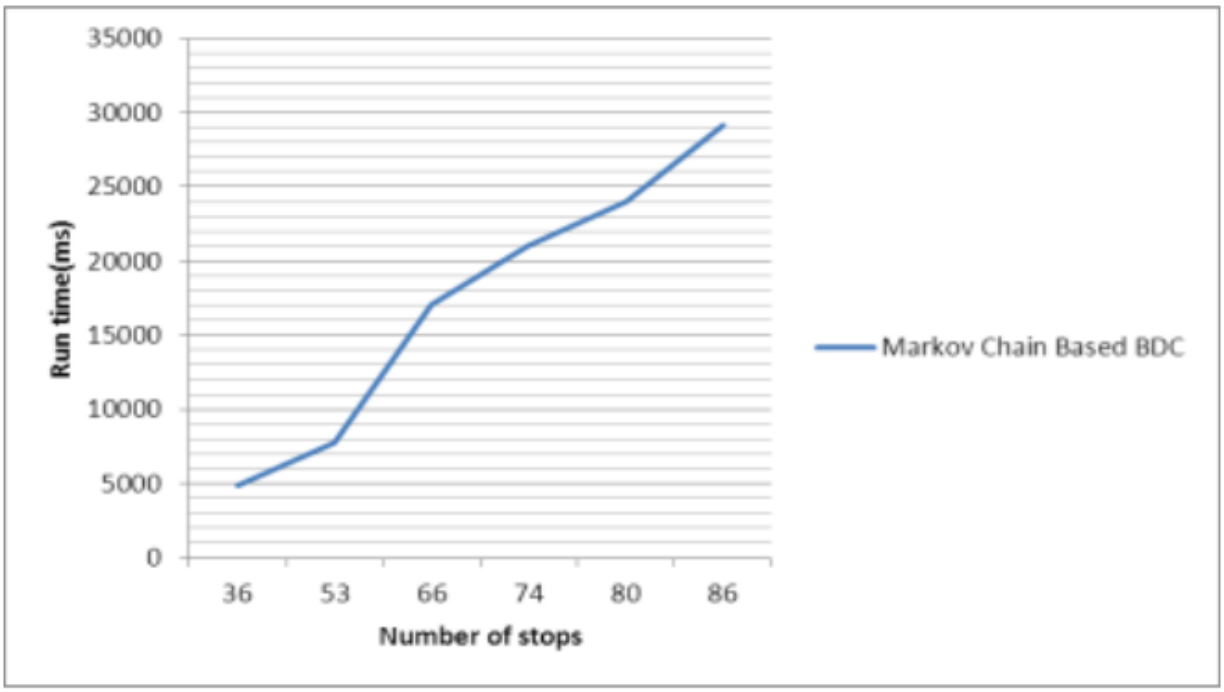


Figure 11. Markov Chain based Bayesian Decision Tree Algorithm Run Time Analysis.

The Markov chain based BDC algorithm can save a significant amount of run time compared with the Basic BDC algorithm. The average performance gains can achieve to 142 times faster than the basic algorithm. This is because most of the redundant calculation steps have been already excluded using Markov chain property.

CONCLUSION

Different from most entry-only AFC systems in other countries, Beijing's AFC system does not record boarding location information when passengers embark 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 proposed a Markov chain based Bayesian decision tree algorithm to infer passengers boarding stops. This algorithm is based on Bayesian inference theory, and the normal distribution of travel speed between adjacent stops is used to depict the randomness of passenger boarding stops. Both the mean and the standard deviation can be obtained from GPS data or distance-based fare routes. Moreover, stationary Markov chain property is also incorporated to further reduce the computational complexity of the algorithm

to a linear load. The optimized algorithm is proven its accuracy using the SC transaction data.

This algorithm can be improved in various ways; for instance, the algorithm does not perform well under the circumstance that the travel speed between adjacent stops is not distinct, i.e., the travel speed probability calculated for each stop is similar. The potential countermeasure for this issue is to incorporate heterogeneity, e.g., the accessibility of a subway station or a central business district (CBD) for each transit stop.

In summary, the Markov chain based Bayesian decision tree algorithm provides both effective and efficient data mining approach for passenger origin data extraction. It sets up a great foundation to mine transit passenger ODs from the SC transaction data for transit system planning and operations.

ACKNOWLEDGMENTS

The authors would like to appreciate the funding support from the National Natural Science Foundation of China (51408019) and the Fundamental Research Funds for the Central Universities. All data used for this study were provided by Beijing Transportation Research Center (BTRC). We are grateful to BTRC for their data supports.

REFERENCES

Bayes, Thomas; Price, Mr. An essay towards solving a problem in the coctrine of chances, *Philosophical Transactions of the Royal Society of London* 53 (0): 370–418, 1763.

Beijing Transportation Research Center, Beijing transportation smart card usage survey, Research Report. 2010.

Beijing Transportation Research Center, the 4th Beijing Comprehensive Transport Survey Summary Report, Jan. 2012.

Chen, J., 2009. Research on travel demand analysis of urban public transportation based on smart card data information, Ph.D. dissertation, Tongji University.

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, Vol. 2063, 2008, pp. 63-72.

Chu, K.K. and Chapleau, R. Augmenting transit trip characterization and travel behavior comprehension: multiday location-stamped smart card transactions. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2183, Transportation Research Board of the National Academies, Washington, DC, 2010, pp.29–40.

Cooper, G. F., The computational complexity of probabilistic inference using Bayesian belief networks, *Artificial Intelligence*, Vol. 42, 1990, pp. 393-405.

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*, Vol. 2072, 2008, pp. 30-37.

Federal Highway Administration, Travel Time Reliabiliy: Making it there on time, all the itme, 2006. Accessed on line at: http://ops.fhwa.dot.gov /publications/tt_reliability/, on Apr. 18th, 2013.

Furth, P. G., Hemily, B., Muller, T. H. J., and Strathman, J. G., TCRP report 113: Using archived AVL-APC data to improve transit performance and management, Transportation Research Board, 2006.

Gao, L.X. and Wu, J. P., An algorithm for mining passenger flow information from smart card data, *Journal of Beijing University of Posts and Telecommunications*, Jun. 2011, vol. 34, No.3, 2011, pp. 94-97.

ICF consulting, Center for urban transportation research, Nelson/Nygaard, ESTC. Strategies for Increasing the Effectiveness of Commuter Benefits Programs. TCRP report 87, Transportation Research Board, 2003.

Jang, W, Travel time and transfer analysis using transit smart card data, *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2144, 2010, pp.142–149.

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*, Vol. 175, Issue 1, 2006, pp. 16-34.

Kittelson & Associates, Inc., Urbitran, Inc. LKC Consulting Services, Inc., Morpace International, Inc., Queensland University of Technology, and Nakanishi, Y., TCRP Report 88, A guidebook for developing a transit performacne-measurement system, Transportation Research Board, National Research Council, Washington, D.C., 2003.

Lou, Y., Zhang, C., Zheng, Y., Xie Xing, Wang, W., and Huang, Y., Map-matching for low-sampling-rate GPS trajectories, Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 352-361, 2009.

Ma, X., McCormack, E., Wang, Y., Processing Commercial GPS Data to Develop a Web-Based Truck Performance Measures Program, *Transportation Research Record: Journal of the Transportation Research Board*. Vol.2246, 2011, pp.92-100.

Ma, X., Wang, Y., Feng, C., and Liu, J. Transit smart card data mining for passenger origin information extraction. *Journal of Zhejiang University Science C*,, Vol. 13, No. 10, 2013, pp. 750-760.

McKenzie, B. and Rapino, M. Commuting in United States: 2009, American Community Survey Reports. Accessed on line at: http://www.census.gov /prod/2011pubs/acs-15.pdf, on Oct. 7th, 2012.

Texas Transportation Institute, 2005 urban mobility report, Texas A&M University, 2005.

Zhou, T., Zhai C., and Gao Z., Approaching bus OD matrices based on data reduced from bus IC cards. *Urban Transport of China*, vol. 5, no.3, 2007, pp. 48-52.