MetroTime: Travel Time Decomposition under Stochastic Time Table for Metro Networks

Haengju Lee
DGIST, Korea

Desheng Zhang
Rutgers University, USA

Tian He
University of Minnesota, USA

Sang H. Son
DGIST, Korea

***Abstract*—One of essential components of public transport systems is to provide travel time estimates for a better travel experience. Based on these estimates, travelers can plan their departure time to meet their target time of arrival. Most of existing work has been focused on estimation on passenger riding time, which is relatively stable. However, a significant portion of time for a subway trip is spent on unstable walking and waiting. As a result, the work solely based on riding times underestimates the actual travel times. To fill the gap, we analyze travel data from automated ticketing systems, which are collected from a large group of passengers in a cost-effective way. We estimate each component (i.e., walking, waiting, and riding) of the travel time using tap-in and tap-out records of these passengers, by a novel travel time decomposition. We evaluate the performance of our travel time decomposition method based on large-scale real-world smart card data from more than 2 million users from Chinese city Shenzhen with 15 million smart card records. The results show that our estimation has an average estimation error of 8% on average and outperforms a baseline approach by 38%. Based on our travel time estimates, we further propose a practical application: digital advertising based on up-to-date travel demand.**

***Index Terms*—Stochastic time table, smart card, time decomposition, digital advertising, metro demand model.**

I. Introduction

Recently, the urban transportation systems, e.g., subway, bus and taxi, become more advanced by well-equipped sensing and communication components to improve ridership experiences. Compared with other urban transportation systems, the subway system is more reliable during peak hours, which means that passengers are expected to spend predicted travel times in subway than in buses or taxis, leading to millions of people riding the subway for their daily commutes [7]. In subway services, the travel time estimation is very important for subway passengers in planning departure times to meet their target arrival times. The subway system publishes static or dynamic time tables that help passengers plan their trips [14]. However, the subway systems sometimes fail to run trains punctually based on prefixed schedules due to various events (e.g., maintenance, a switch problem at a station). As a result, these time tables cannot provide accurate and fine-grained travel information, e.g., how long it takes to go from a ticket gantry at a particular station to its platform [14].

Automated fare collection systems (AFC) have been widely adopted in many metropolitan cities around the world, e.g., New York City, Beijing, and Shenzhen [15], [3]. These AFC systems enable us to calculate fine-grained travel time through subway networks in a cost-effective way because (i) the smart card data have been collected already for accounting purposes and no additional process or infrastructures are required; (ii) the fine-grained travel time can be measured by the difference between the time stamps of tap-out and tap-in transactions based on the smart card data.

Many existing methods have been proposed by using AFC data to infer various subway systems or passenger activities [11], [22], [10], including the travel time segmentation [21]. Given a significant portion of travel time is spent on walking and waiting, it is important to estimate the actual total travel time from an origin to a destination in the subway network. However, even with the AFC data, it is extremely challenging to infer fine-grained passenger activities related to subway trips, e.g., (i) how long a passenger spends time on walking from a ticket gantry to a platform, (ii) how long he/she waits on the platform to board a train, (iii) how long he/she spends time on the train, (iv) how long he/she spends time on walking from the platform to a ticket gantry to tap out. This is because the AFC data are mainly collected for accounting purposes, instead of tracking these fine-grained activities.

In this paper, we utilize AFC data to infer these fine-grained passenger activities. In particular, we propose a framework called MetroTime based on the travel time decomposition for metro networks built upon AFC data. We state our main contributions specifically as follows.

- To our knowledge, we perform the first work to optimize the travel time decomposition with large-scale data from metro AFC systems, in contrast to existing works focusing on in-vehicle travel time estimations.
- Based on extensive AFC data, we present an analytical framework for metro networks with three key components: (i) a new travel time decomposition method based on inter-relationships of travel times between stations with lightweight computational costs, whereas the existing work utilizes intensive sorting, mapping, and grouping operations; (ii) a waiting time estimation technique where we provide a confidence interval, instead of a single estimator, in terms of tap-in time to provide proper inferences;
- We evaluate the performance of our framework through an extensive trace-driven evaluation based on a data set from 2 million passengers and 15 million records in Shenzhen, China. We collect riding time reports from the Shenzhen metro website and use them to validate our

978-1-5090-6517-2/17/$31.00 ©2017 IEEE

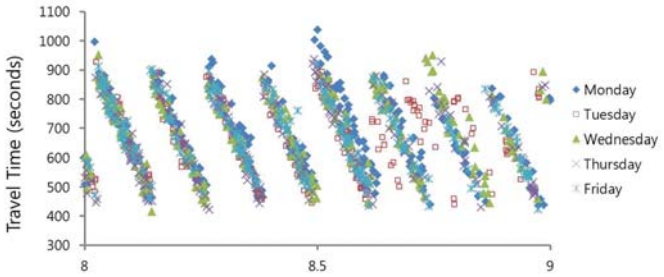


Fig. 1: Travel times (in seconds) between 8 am and 9 am from Yitian station to Futian station on line 3

riding time estimates. The result shows that our method has a 8% estimation error on average. In addition, we show that the proposed travel time estimation can lower the prediction error by as much as 38%, compared with statistical approaches.

- Based on the analytical framework, we design a novel application of advertisement to validate the real-world value of our framework.

The paper is organized as follows. The research motivation is in Section II. We introduce our model and problem in Section III. Sections IV explains how to decompose the travel time. Section V evaluates our method using Shenzhen smart card data. We explain the application in Section VI. Section VII reviews the related work, followed by conclusions in Section VIII.

II. Motivation

We present an interesting application that benefits the public transportation as our research motivation. We also overview the collected smart card data.

A. Application: Digital Advertising

Digital advertising becomes a great success in major metropolitan transit systems. The digital screens are placed inside subway stations to show advertisements to passengers. The metro network sells advertising time of a digital screen to advertisers and charges them based on the length of advertising time. According to the MTA transit system [13], the advertisement revenue has risen dramatically, i.e., from $38 million in 1997 to $130 million in 2013.

The major problem that the system faces is how to estimate the up-to-date travel demand for the spot where a digital screen is installed. The empirical study [20] reveals that the coarse-grained (e.g., one hour) demand is accurately predictable, but the fine-grained (e.g., one minute) demand is unpredictable based on historical data. The fine-grained demand is unpredictable because a train fails to arrive punctually every day. Furthermore, it is unpredictable because a passenger does not arrive at a station at the same time daily.

Because the length of an advertisement is normally less than one minute, it is essential to predict the fine-grained demand accurately. To effectively schedule advertisements for a maximum exposure, we explain the online demand modeling by using real-time tap-in information and the travel time decomposition technology in Section VI.

B. Data Set

Our data set is the metro transaction data from Shenzhen, China. The Shenzhen subway network (serving the mobility of the city) has five lines in 2013 and will have eight more lines over the next seven years.

Each card swiping record includes card ID, station ID, date, time, and tapping in or out. There are 118 metro stations in Shenzhen. The summary of the data collected for our evaluation is in Table I. The data set contains 2,854,022 unique smart cards with 15,466,305 total card transactions ranging from 10/21/2013 (Monday) to 10/25/2013 (Friday) (i.e., five consecutive weekdays).

TABLE I: Data summary and record format

Collection Period	10/21/2013 (Mon.) - 10/25/2013 (Fri.)
Number of Cards	2,854,022
Number of records	15,466,305
Format: Card ID, Station ID, Date & Time, Tap-in/out	

Figure 1 displays the travel times between two stations in terms of tap-in times for those five days. We can roughly say that the longer a passenger waits at the tap-in station, the longer he/she spends on traveling. This simple argument enables us to derive several findings. Trains arrived at the tap-in station regularly for all the five days before 8:30 am. However, it seems that the service was delayed right after 8:30 am on Monday. In addition, it seems that trains arrived earlier after 8:30 am on Tuesday. The irregular service disappears as time approached to 9:00 am. Hence, Figure 1 implies that the trains were operated stochastically.

III. Model and Problem Definition

This section introduces components constituting the metro travel time, proposes an analytical framework, and then specifically states the problem.

A. Activities in Metro Network

We describe activities happening in a typical subway trip. Tapping in to enter an origin station and tapping out to exit a destination station take place at ticket gantries that are typically located away from train platforms. Hence, a typical subway trip is expected to have several activities. Assume that a passenger enters station i, by tapping in his/her smart card at a ticket gantry and walks to the platform. He/she waits on the platform until his/her train departs. After the train departs, he/she spends in-vehicle time to travel to station j. Finally, he/she alights when the train arrives at the platform of station j and walks to a ticket gantry to tap out his/her smart card.

B. Notations

We define notations for those activities. The subway network consists of multiple lines, but we define notations for only one line for simplicity. Hence, we consider all single-leg trips on the chosen line. The main notations are summarized

in Table II, and we provide explanations for them hereinafter.

TABLE II: Notations

Notation	Description
N	Number of stations
T_i^K	Minimum Walking time for a normal person at station i
T_{ij}^R	Riding time from station i to station j
$T_i^W(t)$	Waiting time at station i when tap-in time is t
T_i^D	Dwell time at station i
$T_{ij}(t)$	Travel time from station i to station j when tap-in time is t

The number of stations is denoted by N (station 1, ..., station i, ..., station N). Except for a circular line, which runs clockwise or counter clockwise, the line has two end stations that are connected with only one other station. For a circular line, we randomly select one station to name it station 1 and name the rest stations clockwise.

We consider the walking time from a ticket gantry to the platform and that of the reverse movement (i.e., from the platform to the ticket gantry). If there are many different locations for ticket gantries, which is especially true for a large and complex station, the two routes may be different. Hence, we consider a specific ticket gantry with a minimum distance to the platform and call it the *target gantry*. In this case, the two routes are same, and we make the following assumption.

Assumption 1. *The walking time of a normal passenger from the target gantry to the platform (for the tap-in activity) is similar to the walking time for a normal passenger from the platform to the target gantry (for the tap-out activity) with some small tolerable difference.*

The walking time for a normal passenger between the target gantry and its platform at station i is denoted by T_i^K (i.e., the minimum walking time). The train-riding time from a train departure time at station i to a train arrival time at station j is denoted by T_{ij}^R. Depending on when a passenger taps in the card, the platform waiting time varies. Assume that a passenger arrives at the platform just right before his/her train departs. In this case, the waiting time becomes negligible. In contrast, if he/she just misses the train, he/she needs to wait until the next train departs. In this regard, the waiting time at station i is expressed as a function of tap-in time t, which is $T_i^W(t)$. The train dwells at station i to pick up and drop down passengers. The dwell time at station i is T_i^D.

C. Problem Definition

Summing up all the defined notations, we formulate the travel time from station i to station j,

$$T_{ij}(t) = T_i^K + T_i^W(t) + T_{ij}^R + T_j^K, \tag{1}$$

where t is the tap-in time. Figure 2 illustrates this travel time formulation. The blue cross mark represents the train departure time at station i, the red circular mark represents the train arrival time at station j.

We introduce the second assumption to establish an intuitive analytical framework.

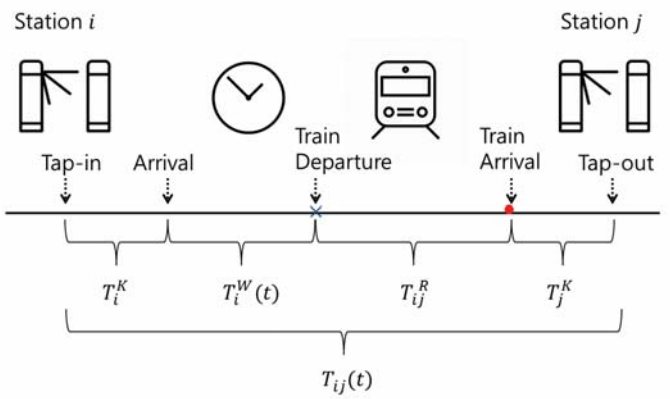


Fig. 2: Activities in a typical subway trip

Assumption 2. *With sufficient trips over a long period of time, at least one passenger boards a train without any waiting to travel from station i to station j using only target gantries.*

Given millions of trips per day in Shenzhen, we consider Assumption 2 holds well. Hence, the travel time of that passenger (without waiting and passing through the target gantries) is the minimum travel time from station i to station j. The time is approximated by taking the minimum over all travel times between two stations. We denote it by T_{ij}^*. Its mathematical formulation is the travel time with zero waiting time as follows.

$$T_{ij}^* = T_i^K + T_{ij}^R + T_j^K. \tag{2}$$

We have multiple unknown variables in Equation (1) and Equation (2). Specifically, we only know $T_{ij}(t)$ and T_{ij}^*, because these can be calculated based on the tap-in times and tap-out times of smart cards. We cannot extract directly when each passenger boards or alights a train. In addition, it is hard to extract how long each passenger spends on walking or riding. To address this challenge, we devote the next section to estimate each component of the travel time in Equation (1).

IV. Network Status Analysis

We estimate each component in Equation (1) and use the hat notation for the estimated value. For example, the estimated value of the walking time is denoted by $\hat{T}_i^K$.

A. Minimum Travel Time

We explain how to estimate the minimum travel time from station i to station j for a normal passenger.

Intuitively the sample minimum (i.e., the smallest observation) among a large number of trips may be applied to estimate the minimum travel time. However, a passenger may run to catch a train. If there are unusual fast-running passengers (i.e., outliers), the sample minimum is not appropriate for the minimum travel time of a normal passenger. To automatically discover outliers, we employ the density based clustering of applications with noise (DBSCAN). The DBSCAN performs well especially in discovering clusters and outliers with arbitrary shaped patterns [5]. After excluding the outliers

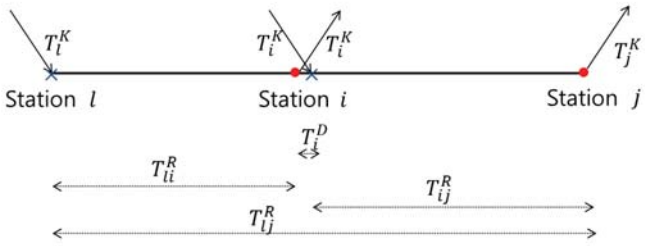


Fig. 3: Relationship of travel times

discovered by DBSCAN, we can compute the minimum value, which is $\hat{T}_{ij}^*$.

B. Walking Time

If station i is not one of end stations (i.e., $i \notin \{1, N\}$), we can find two stations, station l and station j where $l < i$ and $j > i$. We depict the three stations in Figure 3. The components of the minimum travel time formulation in Equation (2) are also displayed. We have the following observation. While the minimum travel time from station l to station i and that from station i to station j include the walking time at station i, the minimum travel time from station l to station j does not. Based on this observation, we have:

$$T_{il}^* + T_{ij}^* - T_{lj}^* = 2 \cdot T_i^K - T_i^D. \tag{3}$$

Note here that our trip data cover all pairs of stations along a single line. For each $i = 2, ..., N-1$, we have $(i-1) \cdot (N-i)$ equations of Equation (3), leading to an over-determined system for T_i^K:

$$T_i^K = \frac{T_{li}^* + T_{ij}^* - T_{lj}^* + T_i^D}{2},$$

where $l = 1, ..., i-1$ and $j = i+1, ..., N$. We derive the least-square solution for this over-determined system (i.e., $(i-1) \cdot (N-i)$ equations for one unknown value). In this case, the least-square solution is simply the average value of them. The average value is $\hat{T}_i^K$ for $i = 2, ..., N-1$.

We do not include the walking time estimation at end stations in this paper because it requires additional information besides the smart card data.

C. Riding Time

We estimate the riding time, T_{ij}^R, from station i to station j. Using the minimum travel time in Equation (2), we can estimate it because we already estimate the walking times at both stations, $\hat{T}_i^K$ and $\hat{T}_j^K$. The estimated riding time is

$$\hat{T}_{ij}^R = \hat{T}_{ij}^* - \hat{T}_i^K - \hat{T}_j^K. \tag{4}$$

D. Waiting Time Inference

Each travel time record in the smart card data, $T_{ij}(t)$, includes the waiting time at station i, whereas the minimum travel time, $\hat{T}_{ij}^*$, does not. This means that we can extract the waiting time information by subtracting the minimum travel time from the travel time. It is noteworthy that $\hat{T}_{ij}^*$ is the estimation for a normal passenger with a normal walking speed. Depending on passengers' walking speeds, there may be variability in waiting times. We explicitly express the variability as follows.

$$T_i^W(t) = T_{ij}(t) - \hat{T}_{ij}^* + \epsilon,$$

where ϵ is the unobservable random part resulting from individual walking characteristics. If we consider only a trip from station i to station j, we may suffer from data sparsity. To resolve the issue, we aggregate waiting time data as long as their tap-in stations are station i. This makes sense because waiting occurs only at tap-in station i. The aggregated data set is

$$\{T_i^W(t) | \forall t\} = \{T_{ij}(t) - \hat{T}_{ij}^* | \forall t \text{ and } i < j \leq N\}.$$

As mentioned before, the waiting time depends on when the passenger taps in the smart card within the train schedule. In this regard, the waiting time with respect to tap-in time t decreases over some time interval (as the train approaches the platform) and then suddenly jumps up at some points (when he/she just misses the train). Hence, the repeating cyclic patterns are expected to be observed. For illustration, we plot waiting times for a station using Shenzhen smart card data as in Figure 4. It is the waiting time with respect to the tap-in time.

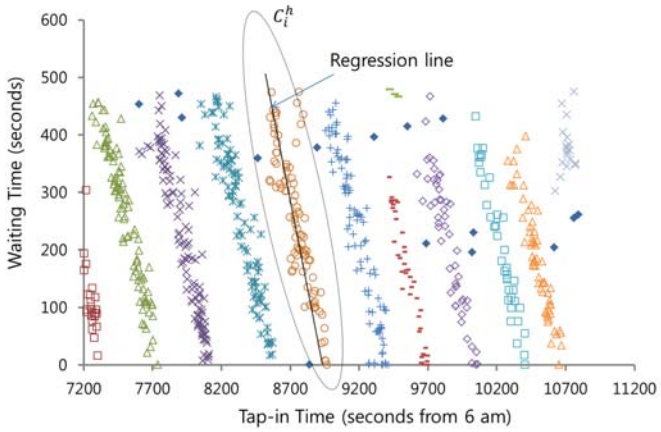


Fig. 4: Cluster result by DBSCAN when tap-in station is Yitian with 10/12/2013 (Monday) data

To automatically discover waiting time patterns, we employ the DBSCAN. The applied DBSCAN result is also shown in Figure 4. The shaded diamond dots represent outlier points. Let clusters of DBSCAN be $C_i^1, C_i^2, ..., C_i^M$, where M is the number of clusters. Note that the outlier points are excluded from this membership. Points in a cluster show a linear relation of the waiting time and the tap-in time, but still have individual variability represented by ϵ. The best line is found by applying the regression analysis to those points.

For each cluster C_i^h, let $\underline{t}_i^h$ be the smallest tap-in time and $\bar{t}_i^h$ be the largest tap-in time. The estimated coefficients are

denoted by a_i^h for the intercept and b_i^h for the slope. Then, if tap-in time t is in C_i^h (i.e., $\underline{t}_i^h \leq t \leq \bar{t}_i^h$), the waiting time estimation is $a_i^h + b_i^h \cdot t$. Time t may not be in any cluster. This may happen during non-peak hours when there are no sufficient data points constituting a cluster. If time t falls in this time frame, the average waiting time is used for its estimation.

$$\hat{T}_i^W(t) = \begin{cases} a_i^h + b_i^h \cdot t & \text{if } \underline{t}_i^h \leq t \leq \bar{t}_i^h \\ \frac{\sum_t T_i^W(t)}{|\{T_i^W(t)|\forall t\}|} & otherwise. \end{cases} \quad (5)$$

The above analysis is for one-day smart card data. Because the train schedule varies daily due to some events, this value needs to be treated as a random variable. The process repeats for all days in the data to have multiple estimations. We let its mean value be $E[\hat{T}_i^W(t)]$ and its standard deviation be $\sigma[\hat{T}_i^W(t)]$. The standard error is $S.E.[\hat{T}_i^W(t)] = \frac{\sigma[\hat{T}_i^W(t)]}{\sqrt{\text{number of days}}}$. Hence, the confidence interval for $\hat{T}_i^W(t)$ is provided, instead of the single estimator.

It is worth to note that the DBSCAN is computationally expensive, but it can be done offline. For the real time application, we note rely on the outcomes of regression analysis.

E. Travel Time Inference

Using the previous waiting time estimate in Equation (5), we infer the travel time from station i to station j at tap-in time t as follows.

$$\hat{T}_{ij}(t) = \hat{T}_i^W(t) + \hat{T}_{ij}^*. \quad (6)$$

Again, the above equation is for one-day smart card data. Similarly, we obtain the confidence interval for $\hat{T}_{ij}(t)$ by obtaining multi-day estimations.

V. Evaluation

In this section, we evaluate the performance of the proposed travel time decomposition method.

A. Walking Time and Riding Time

According to Equation (4), the estimate of the walking time is correlated with that of the riding time. This correlation enables us to indirectly evaluate the walking time estimate by evaluation the riding time estimate. For this, we compare riding time estimates with riding time values posted in the Shenzhen metro website [16]. The indirect evaluation is beneficial because we can avoid the expensive on-site investigation to measure the actual walking times.

Figure 5 is the cumulative distribution function (CDF) of the walking time estimations for all the five lines. Most of estimated walking times are less than 80 seconds. Around 80% of them are less than 33 seconds. By using the walking time estimates, we infer the riding time estimates by using Equation (4). Figure 6 compares the CDF of riding time estimates with the CDF of riding time values posted on the Shenzhen metro website. The two graphs are alined well, which means that the estimates are very close to the true values. In fact, the average estimation error is only 7.65%. Therefore, we can conclude that the decomposition method accurately estimates the riding times, and also the walking times.

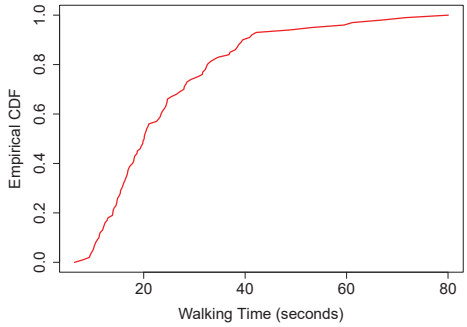


Fig. 5: CDF of walking time estimations

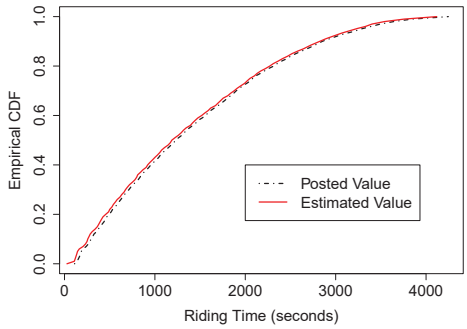


Fig. 6: CDFs of estimations and posted values of riding times

B. Waiting Time and Travel Time

We test the performance of the waiting time and travel time estimations. We call our proposed method as the regression method because the regression analysis is hired for the waiting time estimation.

1) Setup for Evaluation: For comparison, we introduce two alternative methods: the time table method and the average method. The first one is proposed by [21], which relies on the estimation of the train time table. If passengers travel together in the same train, they would get off their destination station together. Based on this idea, the train time table is constructed by clustering the tap-out times. The second one computes the average travel time from a tap-in station to a tap-out station. This is the static estimation method, unlike the other two methods, because the estimate does not depend on tap-in times of passengers.

We set aside the Friday (10/25/2013) data for testing. The remaining data, from Monday (10/21/2013) to Thursday (10/24/2013), are used for training. Note that we have the travel time records as ground truth. The selected performance measure is the mean absolute error (MAE) of the travel time estimate from the ground truth.

2) MAE Comparison: Table III compares the three methods. The MAE of the average method is 68 seconds, 163 seconds for the time table method, and 58 seconds for the regression method. As reported, the simple average method

outperforms the time table method. The proposed regression method improves the MAE value by 15% over the average method. We divide the data set into two groups: peak-hour data and non-peak-hour data. If the tap-in time is in peak hours (7 am to 9 am; 5 pm to 7 pm), it is in peak-hour data. It is noteworthy that whether or not the tap-in time is in peak hours becomes significant for the time table method, compared with the other methods: The MAE is 253 seconds for the peak hour data and 74 seconds for the non-peak hour data.

TABLE III: MAE comparison (unit is in seconds)

Hours	Average	Time Table	Regression	Improvement
Peak	65	253	57	12%
Non peak	71	74	59	17%
Overall	68	163	58	15%

Note: Improvement is the MAE improvement of the regression method over the average method.

We use the ground average value if a data point is not in any DBSCAN cluster (refer to Equation (5)). Hence, data points possessing cluster memberships are truly affected by the regression method. To measure the true impact, we focus on about 75% of the test data with memberships. Table IV compares the average method with the regression method using this partial data set. As expected, higher improvement of 26% is obtained by using the regression method. The reason is that if a data point is not in any cluster, the regression method uses the average value, resulting in no difference between the two methods. This zero difference drags down the overall MAE improvement.

TABLE IV: MAE comparison (unit is in seconds) for data points with membership affected by regression analysis

Hours	Average	Regression	Improvement
Peak	63	52	17%
Non peak	65	41	38%
Overall	64	47	26%

Note: About 75% of the test data have memberships.

We discuss several interesting observations from Table III and Table IV:

- The average method performs better for the peak-hour data than for the non-peak-hour data. When only one estimate is used for trips of a pair of stations, it performs better when trains are operated more frequently. The frequent operation reduces the waiting time variation among passengers. The Shenzhen metro operates trains more frequently during peak hours.
- The regression method performs well for the non-peak-hour data with membership. Passengers usually catch their trains during non-peak hours, resulting in nice patterns for the DBSCAN (in contrast, they may miss trains during peak hours). At the same time, there may not be sufficient data points to constitute a cluster for a certain time frame. That is why the regression method performs well after excluding data points without membership.
- The time table method especially does not perform well for the peak-hour data. We discuss more in the following subsection separately.

3) Over-estimation of Time Table Method: We discuss why the time table method does not perform well especially with the peak-hour data. Figure 7 graphically compares the estimation outcomes for a tap-in station (i.e., Guomao). The graphs are the travel time estimates with respect to the real travel times. The 45 degree line in each graph represents the perfect fit. The average method assigns the same estimate for all the trips having the same tap-out stations. That is why we see many horizontal lines in Figure 7a. Figure 7b is for the regression method, and Figure 7c is for the time table method. By comparing those three graphs, we see that the regression method, having the points located much closer to the 45 degree line, outperforms the other two methods. Many points lying above the 45 degree line in Figure 7c represent over-estimations. These over-estimations result from failing to detect all the train departures through the data processing. During peak hours, trains are operated more frequently. In that case, a long line of passengers, riding on different consecutive trains, are clustered as one group. Thus, it may diagnose one departure during that time span, instead of detecting the actual multiple departures. Figure 7d shows the estimates of the time table method only for the peak-hour data. By comparing it with Figure 7c, we can clearly see that the over-estimations mostly occur in the peak-hour data.

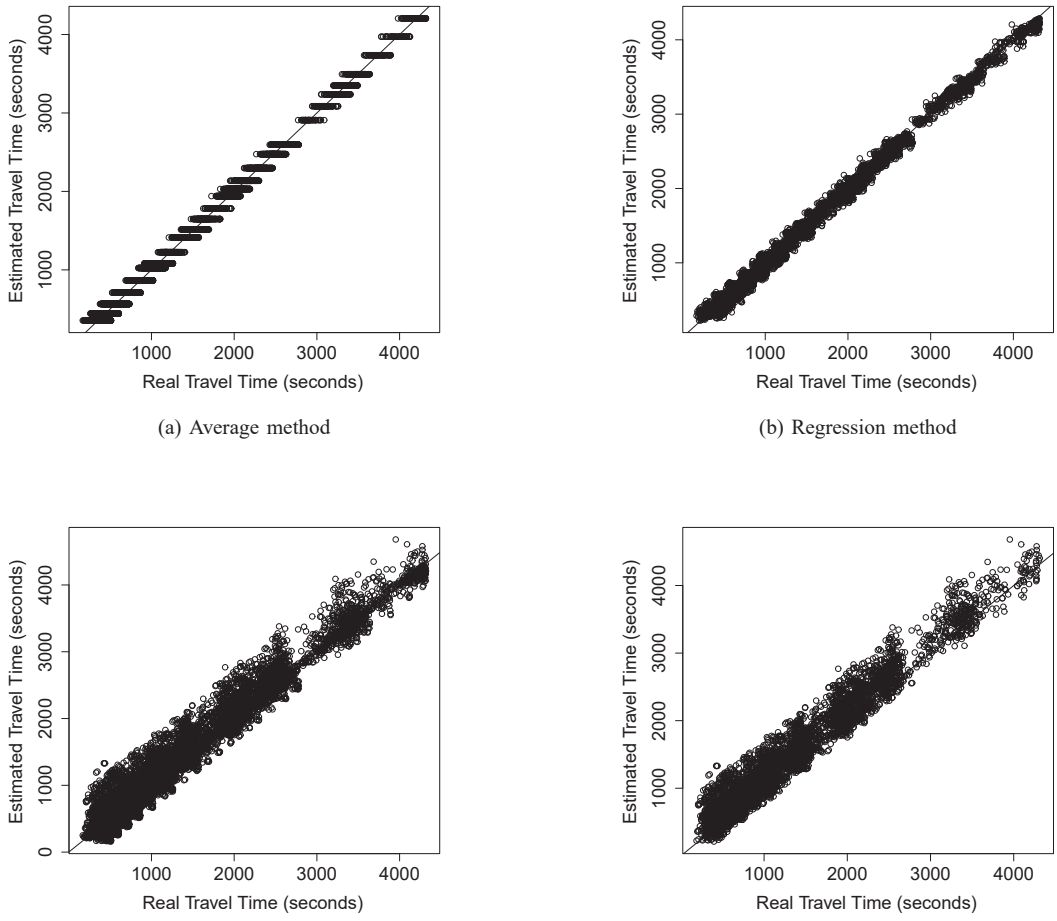
VI. Application: Digital Advertising

We introduce an interesting application based on the two-level demand estimation method. For simplicity, assume that there is only one advertiser that makes a contract to display one-minute advertisement n times in a day at station i. A digital screen is on the platform of station i. What would be the best strategy for the metro advertising system to provide efficient advertising service?

As mentioned in Section II, the coarse-grained demand is accurately predictable. For easier understanding, we use explicit numbers to sketch the application. We use 30 minutes as our coarse-grained time interval, leading to 48 time frames in a day. On the other hand, we use 1 minute as our fine-grained time interval (i.e., time slot), resulting in 30 time slots in a frame. For each historical tap-in record at station i, we consider that the passenger was at the platform after time $\hat{T}_i^k$ and assign the passenger to a corresponding frame. For each tap-out record at station i, we consider the passenger was at the platform before time T_k^i and assign him/her accordingly. We distribute n advertisement slots into 48 time frames that are proportional to the stable historical passenger traffic percentage at station i.

The above frame-level allocation can be done offline. We now explain how to schedule the slot-level allocation online. We focus on one frame to explain the dynamic schedule. Assume that $m \leq 30$ slots are assigned to the frame offline. We have 30 candidate slots and choose m slots from them for advertisement. We let the start time of the frame be τ^s (i.e., the first slot is $[\tau^s, \tau^s + 1]$).

Assume that the current time is $\tau = \tau^s$. For each time slot, we estimate the future passenger demand. If the first slot is (or



(a) Average method

(b) Regression method

(c) Time table method (overall)

(d) Time table method (peak hours)

Fig. 7: Estimation performance for tap-in station of Guomao on Line 1

is not) one of top m heavy traffic slots, then the advertisement is (or is not) displayed. At time $\tau = \tau^s + 1$, we similarly do the same process to schedule the remaining $m-1$ (or m) slots.

Specifically, we explain how to estimate the slot-level demand at time τ online. Assume that passenger θ taps-in his/her card at time $t < \tau$ at station l and is still in the subway network. If $l = i$, then he/she is expected to be on the platform of station i at time $t + \hat{T}_i^K$ and is counted in a slot including that time. If $l \neq i$, we derive the distribution of his/her destination stations using his/her historical card transactions, conditional on tap-in time t and tap-in station l. If the personal data for him/her are not sufficient, we then use the general distribution. From the distribution of the destination stations, we can pinpoint the probability that his/her destination station is i, which is denoted by $\hat{P}_{li}$. The expected tap-out time of passenger θ is $t + \hat{T}_{li}(t)$. Hence, the platform arrival time at station i is estimated to be $t + \hat{T}^{li}(t) - \hat{T}_i^K$. Then, he/she is counted in a slot including that time, but only $\hat{P}_{li}$ (≤ 1), instead of one person, is counted.

VII. Related Work

There is a broad literature mining travel data collected by AFC systems. An excellent literature review including smart card technologies and privacy issues can be found in [18].

AFC data are used to study the performance of overall transportation systems. The transaction data are used for transit demand modeling [4] and for service reliability measures [2], [19]. State transfer trips are analyzed to provide information about passengers' transfer location choices [8].

To plan and design urban facilities and services, it is important to understand the mobility patterns (e.g., daily two peaks during weekdays, relatively even distribution during weekends). For this purpose, the aggregated temporal and spatial patterns are studied using AFC data [11], [22], [12].

The above works aggregate the trip data to derive system-performance measures and to obtain traffic patterns. The other line of research uses AFC data to reveal individual behaviors of passengers. After demonstration of travel time differences among passengers using AFC data, the personalized travel

time estimation is built based on individual travel behaviors [10]. Crowd levels are predicted to provide more personalized travel plans [1]. By avoiding the overcrowded times, the quality-based planning service can be provided. After extracting individual travel behaviors, a tool can be built to recommend the best fare to purchase [9]. In the sense that we focus on each individual trip transaction for the travel time decomposition, our work is related with this line of research.

The travel time decomposition has been studied [6], [17], [21]. The waiting time is estimated by integrating the AFC data with published time tables [6]. However, we obtain it without the aid of the unreliable (due to unexpected events) time table. The uniform walking time and dwell time across every station are assumed in [17]. In addition, the method by [17] requires the physical distance among stations.

Our work is close to [21]. However, we state the major differences. First, the decomposition method in [21] requires intensive sorting, mapping, and grouping operations to search for passengers without waiting (called boarder-walkers). Thus, our method has a potential advantage in saving computational efforts. Second, the method by [21] does not work for end stations because it requires two boarder walkers (i.e., a boarder walker for tapping in and another boarder walker for tapping out) for a chosen direction. Note that if a direction is chosen, there is only one kind of smart card transaction (either tap-in or tap-out) for the end stations. However, the analysis for end stations can be extended easily (we do not include the analysis in the paper). Therefore, our method is more general. Third, we propose a new method of providing the confidence interval for the travel time estimation.

VIII. Conclusion

This paper addresses the travel time decomposition problem by using the tap-in and tap-out records in AFC data. The decomposition enables efficient digital advertising that is based on the up-to-date travel demand for the spot where a digital screen is installed. The Bayesian demand model is constructed by using real-time tap-in information, passengers' moving patterns, and the travel time decomposition. Furthermore, the constructed stochastic time table enables new personalized applications such as the reliable tap-in time recommendation. A passenger can catch a target train with a given probability by using the recommendation, which will be developed for our future research. It is also an interesting research direction to pursue a user study in order to determine the impact of the new application on travel patterns. In other words, it is important to shed light on how passengers adjust their travel behaviors in response to the new personalized applications. Passengers may arrive in bursty patterns resulting in long queueing lines to tap-in their cards. Hence, queueing theory is applied to add the pre-gantry queuing time to the total travel time. We leave this as an interesting future work.

Acknowledgment

This work was supported by US NSF Grants CNS-1446640 and CNS-1544887 and Global Research Laboratory Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT Future Planning(NRF-2013K1A1A2A02078326).

References

[1] I. Ceapa, C. Smith, and L. Capra, *Avoiding the crowds: Understanding tube station congestion patterns from trip data*, In Proceedings of the 1st ACM SIGKDD International Workshop on Urban Computing, 134–141, 2012.

[2] J. Chan, *Rail transit OD matrix estimation and journey time reliability metrics using automated fare data*, Master's thesis, MIT, Department of Civil and Environmental Engineering, 2007.

[3] China Automatic Fare Fare Collection (AFC) System Industry Report, http://www.prnewswire.com/news-releases/china-automatic-fare-collection-afc-system-industry-report-254141821.html.

[4] K. Chu and R. Chapleau, *Enriching archived smart card transaction data for transit demand modeling*, Transportation Research Record: Journal of the Transportation Research Board, 63–72, 2008.

[5] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, *A density based algorithm for discovering clusters in large spatial databases with noise*, In Proceedings of KDD-96, 226–231, 1996.

[6] M. Frumin and J. Zhao, *Analyzing passenger incidence behavior in heterogeneous transit services using smartcard data and schedule-based assignment*, Transportation Research Record: Journal of the Transportation Research Board, 52–60, 2012.

[7] Introduction to Subway Ridership, http://web.mta.info/nyct/facts/ridership.

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

[9] N. Lathia and L. Capra, *Mining mobility data to minimise travellers' spending on public transport*, In Proceedings of the 17th ACM SIGKDD Conference of Knowledge Discovery and Data Mining, 1181–1189, 2011.

[10] N. Lathia, J. Froehlich, and L. Capra, *Mining public transport usage for personalised intelligent transport systems*, In Proceedings of the 10th IEEE International Conference on Data Mining, 887–892, 2010.

[11] L. Liu, A. Biderman, and R. Carlo, *Urban mobility landscape: real time monitoring of urban mobility patterns*, In Proceedings of the 11th International Conference on Computers in Urban Planning and Urban Management (CUPUM), 1–16, 2009.

[12] L. Liu, A. Hou, A. Biderman, C. Ratti, and J. Chen, *Understanding individual and collective mobility patterns from smart card records: A case study in Shenzhen*, In Proceedings of the 12th IEEE ITSC, 1–6, 2009.

[13] New Screens Bring Added Revenue to the MTA, http://www.mta.info/news/2014/03/13/new-screens-bring-added-revenue-mta.

[14] NYC Subway Time, http://apps.mta.info/traintime.

[15] Automated-Card System Chosen to Collect Fares in New York, http://www.nytimes.com/1991/03/16/nyregion/automated-card-system-chosen-to-collect-fares-in-new-york.html.

[16] Shenzhen Metro Website, http://www.szmc.net/page/en/index.html.

[17] L. Sun, D. Lee, A. Erath, and X. Huang, *Using smart card data to extract passenger's spatio-temporal density and train's trajectory of MRT system*, In Proceedings of the ACM SIGKDD International Workshop on Urban Computing, 142–148,2012

[18] M. Pelletier, and M. Trépanier, and C. Mrency, *Smart card data use in public transit: A literature review*, Transportation Research Part B, 39, 119–140, 2010.

[19] D. L. Uniman, J. Attanucci, R. G. Mishalani, and N. H. M. Wilson, *Service reliability measurement using automated fare card data*, Transportation Research Record: Journal of the Transportation Research Board, 92–99, 2010.

[20] D. Zhang, R. Jiang, S. Wang, Y. Zhu, B. Yang, T. He, and J. Cao, *Everyone Counts: Data-Driven Digital Advertising with Uncertain Demand Model in Metro Networks*, In Proceedings of the IEEE Big Data, 898–907, 2015.

[21] F. Zhang, J. Zhao, C. Tian, C. Xu, X. Liu, and L. Rao, *Spatiotemporal segmentation of metro trips using smart card data*, IEEE Transactions on Vehicular Technology, 65, 1137–1149, 2016.

[22] J. Zhao, C. Tian, F. Zhang, C. Xu, and S. Feng, *Understanding temporal and spatial travel patterns of individual passengers by mining smart card data*, In Proceedings of the IEEE 17th ITSC, 2991–2997, 2014.