Knowl Inf Syst (2015) 44:299–323
DOI 10.1007/s10115-014-0763-x

REGULAR PAPER

Reconstructing individual mobility from smart card transactions: a collaborative space alignment approach

Fuzheng Zhang · Nicholas Jing Yuan · Yingzi Wang · Xing Xie

Received: 12 December 2013 / Revised: 5 May 2014 / Accepted: 3 July 2014 /
Published online: 23 July 2014

Abstract Smart card transactions capture rich information of human mobility and urban dynamics and therefore are of particular interest to urban planners and location-based service providers. However, since most transaction systems are only designated for billing purpose, typically, fine-grained location information, such as the exact boarding and alighting stops of a bus trip, is only partially or not available at all, which blocks deep exploitation of this rich and valuable data at individual level. This paper presents a *collaborative space alignment* framework to reconstruct individual mobility history from a metropolitan-scale smart card transaction dataset. Specifically, we show that by delicately aligning the monetary space and geospatial space with the temporal space, we are able to extrapolate a series of critical domain-specific constraints. Later, these constraints are naturally incorporated into a semi-supervised conditional random field (CRF) to infer the exact boarding and alighting stops of all transit routes, where the features of the CRF model consist of not only pre-defined indicator features extracted from individual trips but also latent features crafted from different users' trips using collaborative filtering. Here, we consider two types of collaborative features: (1) the similarity in terms of users' choices of bus lines and (2) latent temporal patterns of users' commuting behaviors. Extensive experimental results show that our approach achieves a high accuracy, e.g., given only 10 % trips with known alighting/boarding stops, and we successfully inferred more than 79 % alighting and boarding stops from all unlabeled trips. In particular,

F. Zhang · Y. Wang
University of Science and Technology of China, Hefei, China

F. Zhang
e-mail: zhfzh@mail.ustc.edu.cn

Y. Wang
e-mail: yingzi@mail.ustc.edu.cn

N. J. Yuan (✉) · X. Xie
Microsoft Research, Beijing, China
e-mail: nicholas.yuan@microsoft.com

X. Xie
e-mail: xing.xie@microsoft.com

we validated that the extracted collaborative features significantly contribute to the accuracy of our model. In addition, we have demonstrated that by applying our approach to enrich the data, the performance of a conventional method for identifying users' home and work places can be dramatically improved (with 83 % improvement on home detection and 38 % improvement on work place detection). The proposed method offers the possibility to mine individual mobility from common public transit transactions, and showcases how uncertain data can be leveraged with domain knowledge and constraints, to support cross-application data-mining tasks.

Keywords Smart card · Space alignment · Mobility · Collaborative filtering

1 Introduction

Many data-mining tasks benefit from cross-application datasets. Such cases often follow a simple paradigm as illustrated in Fig. 1: The source application generates enormous data, which intends to serve its own needs, but might also be significantly valuable to another target application, where data are limited or not easy to obtain. To name a few, taxi trajectories collected for security management can be leveraged to probe traffic flows [30]. Yet users' search queries can be employed to accurately detect pandemic influenza trends [11].

Mining smart card transactions gives another example that fall in this scope. Smart cards (such as credit cards, fuel cards,[1] campus cards, and public transit cards) facilitate millions of people for digital payment and public transport ticketing in many metropolises. Examples include London's Oyster Card,[2] San Fransisco's Clipper Card,[3] and Beijing's BMAC Card.[4] Overwhelming amounts of transaction data are cumulated in the fare systems every day. Such data are of particular interest to urban planners and location-based service providers, since it reveals urban dynamics and human mobility patterns. Several attempts have been made in mining smart card transactions and show promising prospects in various applications such as mobility modeling [18] and personalized recommendations [16,17].

However, most existing approaches in mining transactions of public transit smart cards suffer from the data uncertainty and incompleteness problems. This is also a challenge to a broad range of compelling applications dealing with cross-application datasets: Data generated from the source application often lack information that is necessary to the target application, e.g., the public transit transactions sometimes do not include the information of trip destinations (the fare does not depend on the destinations thus there is no intention to record such data [16]), but knowing both the origin and destination of a trip is crucial for mining mobility patterns. As a consequence, a considerable amount of work either excludes these uncertain bus trips [17] or focuses on mining aggregated level instead of individual level patterns for uncertain bus trips [18]. A few existing methods have been proposed to recover public transit trips [7,26,29], but most of them assume that at least the origin or the destination is given for each trip, which is sometimes not the case.

To address this challenge, this paper provides a systematic solution to reconstruct fine-grained mobility history at individual level from common smart card transactions, which exemplifies how the data coming from the source application can be enriched in terms of

[1] https://en.wikipedia.org/wiki/Smart_card.

[2] https://oyster.tfl.gov.uk/oyster/entry.do.

[3] https://www.clippercard.com.

[4] http://www.bmac.com.cn.

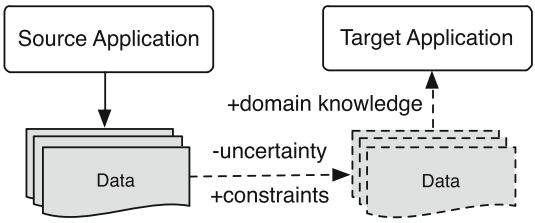


Fig. 1 The paradigm of mining cross-application data

granularity and availability to facilitate the target application. Typically, the data coming from the source application are uncertain yet constrained—take as an example the smart card transactions—such constraints are not only related to the *geospatial* space (e.g., the distance that a passenger walks when transferring bus lines) but may also be implied in the *monetary* space (e.g., the balance of a card), or the *temporal* space (e.g., the time interval between two transits). As shown in Fig. 1, the proposed solution reduces the uncertainty of the data generated from the source application (here, smart card transactions), by incorporating constraints implied in the source application as well as domain knowledge in the target application (mobility mining).

Given the constraints derived by aligning three spaces, we transform the problem of reconstructing individual mobility to a sequential labeling problem. To tackle this problem, we utilize a semi-supervised conditional random field leveraging the obtained constrains. We feed the model with features considering both characters of individual trips and latent factors learned using a collaborative filtering approach. These latent features include two categories: (1) User Interest Features, which incorporate user similarity, e.g., two users who live/work at places close to each other may frequently choose the same bus lines for commuting; (2) Temporal Interest Features, which model temporal patterns of users' commuting behaviors at different time slots (e.g., morning/evening rush hours) or on different days of week (weekdays/weekends).

To the best of our knowledge, this is the first solution that can recover individual bus trips from course-grained smart card data, where the information of both the boarding and alighting stops may be unavailable. In summary, this paper mainly offers the following contributions:

- We have derived a space alignment framework that coalesces the monetary, temporal, and geospatial spaces, to segment all the trips and extract domain-specific constraints, which significantly reduce the number of candidate bus stops, even without the information of the boarding or alighting stops.
- We have applied a conditional random field-based sequential model to infer the actual alighting and boarding stops for each trip, where the well-designed model features such as users' common preferences of choosing bus lines and constraint functions are naturally incorporated, and the known bus stops for some trips are leveraged for training the model in a semi-supervised way.
- We conducted extensive experiments to validate the proposed method with a large-scale human-labeled dataset as ground truth. The experimental results as well as the demonstrations validate the effectiveness of our method.

2 Data

This section explicitly describes the smart card transaction dataset we used in this study, which consists of two tables: the expense records and the charging records, as illustrated in

Table 1 Expense records and charging records

CardID	Bus	Boarding	Alighting	Time	Expense	Balance
(a) Expense records						
1	$N2$	–	–	2013-03-14 09:02	0.8	12.3
2	$L3$	31	19	2013-03-14 17:45	0.4	32.2
3	$N1$	–	–	2013-03-15 08:45	0.4	10.6
3	$L1$	04	22	2013-03-16 18:20	0.8	49.8
CardID				Time	Amount	Balance
(b) Charging records						
3				2013-03-15 18:05	50.0	50.6
4				2013-03-15 18:05	20.0	21.6
5				2013-03-15 18:07	20.0	20.4
6				2013-03-15 18:08	30.0	40.8

Table 1(a) and (b), respectively. The dataset covers a population of 701,250 card holders. We note that the smart card is not limited to the payment for bus transit, but can also be used for other types of payments in this city, such as taxis, subways, and shopping. However, the dataset we obtained only covers bus-related expense records (but the charging records are fully available).

2.1 The expense records

This table contains in total 22.03 million bus-trip records, during the period from Aug. 2012 to May. 2013. Each trip is shown as a row in Table 1(a), containing the following columns:

- **CardID**: the ID of a smart card, where each card has a unique ID, and typically an individual has only one smart card. Note that in this work, all CardIDs are **anonymized** and are not associated with any personally identifiable information or profiles, for protecting users' privacy.
- **Bus**: the line number (encoded by us from the original names) of a bus, where there are two types of bus lines as shown in Table 1(a): **non-ladder-fare** lines (beginning with "N") such as $N2$ and $N1$, and **ladder-fare** lines (beginning with "L") such as $L1$ and $L3$. If you take a non-ladder-fare bus, the fare is identical for the whole line regardless of where you get on or get off the bus, and thus, you are only required to swipe the smart card once you get on the bus. Yet for ladder-fare bus lines, since the fare is calculated according to the distance between the boarding and alighting stops, you have to swipe the card twice: one swipe at the boarding stop, and the other at the alighting stop.
- **Boarding** and **Alighting**: the codes of the boarding and alighting stops. This information is only available for ladder-fare lines, since the fare of the non-ladder-fare lines is fixed as mentioned above, thus the public transport authority does not record the boarding nor alighting stops in the billing system. We note that even for ladder-fare lines, the recorded information is only a code of the bus stop, which identifies how long (in kilometers) the bus stop is apart from the bus stop with the code 0, where the 0-coded stop is unknown to us. That means, the direction of a bus line is not observable, since either the departure stop or the terminal stop of a bus line can be coded with 0.

- **Time**: the exact time that the fee of a bus trip is deducted from the smart card, which also depends on whether it is a ladder-fare line: For non-ladder-fare lines, the recorded time is the moment that you swipe the smart card when you get on the bus at the boarding stop, while for ladder-fare lines, it is the moment before you get off the bus at the alighting stop.
- **Expense**: the expense of a trip. For non-ladder-fare lines, it is a fixed amount, but for ladder-fare lines, it varies according to the distance between the boarding stop and the alighting stop, which can be calculated directly from the boarding column and the alighting column in the table, e.g.,

$$e = a + b \cdot \max(|\text{boarding} - \text{alighting}| - c, 0), \tag{1}$$

  where e is the expense, and a, b, c are system parameters varied for different bus lines. It follows that if the distance between boarding and alighting stops is less than or equal to c kilometers, you should pay a; otherwise, you should pay additional b for every extra kilometer. In such a way, the whole fare system considering all possible boarding/alighting stops looks like a "ladder" (that's the reason it is called ladder fare).
- **Balance**: the remaining balance of the smart card after a trip.

2.2 The charging records

To maintain the usage of a smart card, people typically recharge it when necessary. The charging record, as exemplified in Table 1(b), has a simple schema and is easy to understand. Our dataset contains 5.93 million charging records, each of which includes the columns of **CardID**, **Time** (the time you charge your smart card), **Amount** (how much you charge this time), and **Balance** (how much you have in the smart card after this charging).

2.3 Road network

The road network G is a directed graph $G = (V, E)$, where V is a set of nodes, representing the terminal points of road segments, and E is a set of road segments. Each road segment $e \in E$ contains the information of limit (maximum) driving speed. The road network we used contains 148,110 nodes and 196,307 road segments.

2.4 Data denoising and data labeling

Through a public map API,[5] we can search for the geo-coordinates of the bus stops of each bus line in this city, as well as its pricing information (i.e., we obtain the parameters in Eq. (1) for the bus line), by providing the original bus line names in the expense record. Nevertheless, there are still some bus lines that we failed to find pricing information (95 lines) or the bus stop geo-coordinates (488 lines), since sometimes the name is not complete or ambiguous. Fortunately, these bus lines only account for a small part of all the expense records (about 20 %) in the transaction data, as shown in the statistics of Table 2. Therefore, we removed the records associated with these "unknown" bus lines in our study.

In addition, we conducted a data-labeling program recruiting 102 selected participants to label the specified most frequent ladder-fare lines. In this 4-month program (from Dec. 2012 to Mar. 2013), each participant was provided with a free smart card, and her expense of daily bus transit was reimbursed for labeling the data. Specifically, after signing a consent form regarding the privacy and legal issues, each participant was required to manually record her

[5] http://api.amap.com.

Table 2 Statistics of bus lines and expense records

Line type	#lines	Ratio of records (%)
Lines without coordinates	95	4.16
Lines without price info	488	16.84
Non-ladder-fare	270	36.62
Labeled ladder-fare	124	26.54
Unlabeled ladder-fare	288	15.85

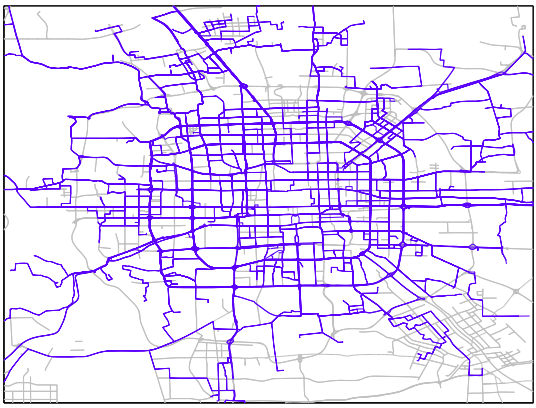


Fig. 2 Labeled ladder-fare bus lines

every trip paid by the smart card during the 4 months, and label the corresponding expense records [as shown in Table 1(a)] in the transaction data, indicating the names of the boarding and alighting stops, as well as the boarding and alighting time.

As described in Sect. 2.1, knowing the direction of a ladder-fare line is equivalent to knowing the mapping between codes and the real names (thus the locations) of the stops. We term these ladder-fare lines as **labeled lines**. If a trip recorded in the expense record belongs to a labeled line, the alighting/boarding stops are called **labeled stops** of this **labeled trip**. As a result, we find out the directions of 124 most frequent ladder-fare bus lines, as shown in Table 2. The labeled lines cover more than 26 % of all trips recorded in the expense records and account for more than 62 % ladder-fare trips (recall that alighting and boarding stops for non-ladder-fare lines are not even recorded in the raw transaction data). Figure 2 plots all the labeled bus lines (colored blue), where the underlying road networks (colored gray) delineate the urban area of this city. Clearly, the labeled lines cover a majority of the urban area. However, our main challenge is how to leverage these partially labeled trips to recover the rest (and much more) unlabeled trips.

3 Methodology

There are three parallel spaces in the transaction data: the **monetary space** $\mathcal{M}$, the **temporal space** $\mathcal{T}$, and the **geospatial space** $\mathcal{S}$.

The balance, charging amount, and expense of a trip are associated with the monetary space, for a given smart card. As shown in the above line of Fig. 3, the balance of a user's smart card rises after the user charges the card, and declines after a trip, where the time

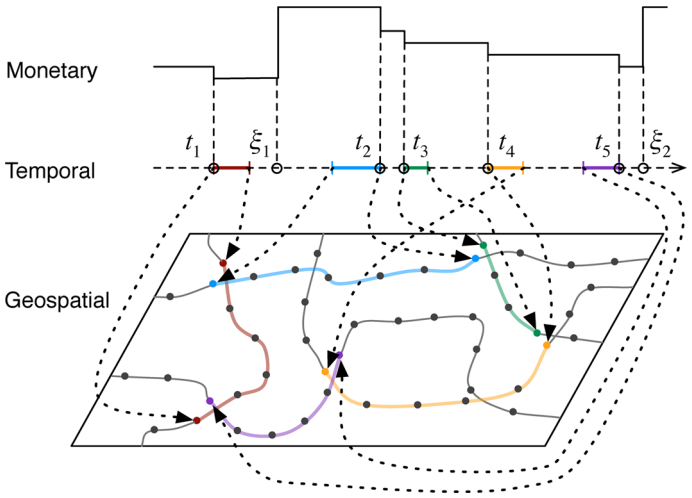


Fig. 3 Space alignment

stamps of expense and charging are points in the temporal space (shown in the middle line of Fig. 3). As described in Sect. 2.1, for non-ladder-fare trips, the time stamps reflect the boarding time, while for ladder-fare trips, they represent the alighting time.

For a certain trip (the timeslot of each trip is denoted as a colored solid line in the middle of Fig. 3), each intermediate point in the temporal space is aligned with a spatial point located in the geospatial space, restricted by the bus line of this trip. In particular, the boarding and alighting time stamps can be mapped to the boarding and alighting stops respectively. Note that the doted lines in the temporal space denote that the time flows, but no bus trips are recorded, e.g., when the users stay at home during night or work at office during daytime.

By superimposing the three spaces, the goal of recovering individual bus trips can be generally described as: *To identify the mapping from the temporal space $\mathcal{T}$ to the geospatial space $\mathcal{G}$ for each trip recorded in the expense records $\mathbb{E}$, given the charging records $\mathbb{C}$ in the monetary space $\mathcal{M}$ and a specified CardID.*

3.1 Preliminary

If the smart cards can only be used for bus trips, then we can continuously track a user's balance without any discontinuous points. However, it is not the case for our data. In fact, the card holders can use their smart cards for other payments, such as taking a taxi or a private car or even shopping. Though we have the full information of the charging records, the expense records only include bus-related expense. Meanwhile, the "dirty" bus lines (without information of price or coordinates) are removed from our data as described in Sect. 2.4. In order to make sure that the consecutive records are really two consecutive bus trips with known price information and coordinates (see Sect. 2.4), we partition an individual's expense records into segments (which is essential for our modeling later), defined as below.

Definition 1 (*Segment*) Let $R = \{r_1, r_2, \ldots, r_n\}$ denote the expense records for a given smart card, with time stamps $t_1, t_2, \ldots, t_n \in \mathcal{T}$ sorted in the chronological order, and expense $e_1, e_2, \ldots, e_n \in \mathcal{M}$ (as recorded in the expense records). Let $\xi_1, \xi_2 \ldots, \xi_m$ be the time stamps when the smart card is charged with amount $c(\xi_1), c(\xi_2), \ldots, c(\xi_m) \in \mathcal{M}$. Let $b(t)$ be the

Algorithm 1: Segmentation

Input: CardId d, expense records $\mathbb{E}$, and charging records $\mathbb{C}$
Output: Segments $\mathbf{S}$

$\mathbf{I} \leftarrow \{1\}$; /* $\mathbf{I} = \{I_i\}_{i=1}^{|\mathbf{I}|}$ is the index of split points */
$\mathbf{E} \leftarrow$ select * from $\mathbb{E}$ where CardID=d order by time; /* $\mathbf{E} = \{E_i\}_{i=1}^{|\mathbf{E}|}$ */
$\mathbf{C} \leftarrow$ select * from $\mathbb{C}$ where CardID=d order by time; /* $\mathbf{C} = \{C_j\}_{j=1}^{|\mathbf{C}|}$ */
$c_i \leftarrow 0, i = 1, 2, \ldots, |\mathbf{E}|$;
$i \leftarrow 1, j \leftarrow 1$;
while $i \leq |\mathbf{E}| - 1$ **do**
  if $j \leq |\mathbf{C}|$ **and** $t_1 < \xi_j < t_i$ **then**
    $c_i \leftarrow c_i + c(\xi_j)$;
    /* $c(\xi_j)$ can be directly read from C_j */
    $j \leftarrow j + 1$;
  else
    $i \leftarrow i + 1$;
    if $b_i + e_i \neq b_{i-1} + c_{i-1}$ **then** /* b_i and e_i can be directly read from E_i */
      $\mathbf{I} \leftarrow \mathbf{I}.add(i)$;
return $\mathbf{S} = \{S_k\}_{k=1}^{|\mathbf{S}|}$, where $S_k = \{E_{i_k}\}_{i_k=I_k}^{I_{k+1}-1}$

balance of the smart card at time $t \in \mathcal{T}$. R is called a **segment** if the following condition holds for $1 \leq i \leq n-1$:

$$b_{i+1} + e_{i+1} = b_i + c_i, \tag{2}$$

where b_i is the balance of the smart card right after the ith trip,[6] i.e., $b_i = \lim_{t \to t_i^+} b(t)$, and

$$c_i = \sum_{\substack{t_i < \xi_j < t_{i+1} \\ j=1,2,\ldots,m}} c(\xi_j), \tag{3}$$

which denotes the total charges between the ith and the $(i+1)$th trip.

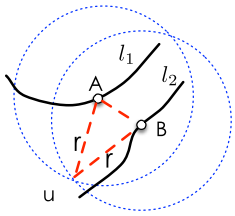
Intuitively, a segment is a sequence of expense records where the balance of the card can be continuously tracked without any missing expense records. Note that we do not explicitly segment the records to days as done in many existing approaches [7,26]; instead, a segment can contain many days and nights as long as no missing points are found. Based on Definition 1, we propose an algorithm to perform the segmentation for a certain smart card with CardID d, as presented in Algorithm 1. This algorithm incrementally calculates c_i defined in Eq. (3) for each record of d, and checks whether condition (2) holds in place. Given the expense records $\mathbf{E}$ and charging records $\mathbf{C}$ of a certain card in the chronological order, the segmentation is obtained in $O(|\mathbf{E}| + |\mathbf{C}|)$ time.

3.2 Constraints for transitions

Recall that in our data, there are non-ladder-fare trips and ladder-fare trips, where the directions are known only to part of the ladder-fare trips (by labeling, as described in Sect. 2.4). Let $S = \{l_1, l_2, \ldots, l_m\}$ denote a segment with m trips in chronological order, where $l_i = (o_i, d_i)$ is the origin (i.e., boarding stop) and destination (i.e., alighting stop) of the ith trip. Assume that all the trips in S are not labeled (without any information of the directions), and l_i has n_i bus stops. Assuming the alighting stop is different from the boarding stop for a trip,

[6] We take the right limit here since $b(t)$ is a step function as depicted on the top part of Fig. 3.

Fig. 4 Proximity constraints



it follows that in the worst case, and there are in total $n_i(n_i - 1)$ possible trips (pairs of boarding-alighting stops) for l_i. Thus, we have

$$\prod_{i=1}^{m} n_i(n_i - 1) \tag{4}$$

candidates for S.

However, by considering several constraints in the monetary space, temporal space, and geospatial space, we can exert several constraints to dramatically reduce the number of candidate trips, even if all the trips are not labeled. In fact, there are two types of transitions (displacements in the geospatial space) in a segment, defined as follows:

Definition 2 (*Inner-transition and outer-transition*) Given a segment $S = \{l_1, l_2, \ldots, l_m\}$ where l_i is a bus trip from boarding stop o_i to alighting stop d_i, we call each transition $o_i \rightarrow d_i$ an inner-transition, where the movement of a user is strictly restricted by the bus (along the bus line). We call each transition between two consecutive trips, i.e., $d_i \rightarrow o_{i+1}$, an outer-transition.

We introduce the constraints for both of the two transitions.

- *Proximity constraints [for outer-transitions]*

Given the limits of walking speed and walking duration, as well as the highly developed transportation systems in metropolises, a citizen's walking scope is usually limited. This is also well supported by results in existing literatures, for example, [3] reported that nowadays American adults walk on average 5119 steps (about 2.5 miles) per day. Another recent study showed that more than 97.6 % walking trips of Sydney citizens are less than 2 km [8], using 3 years data of Sydney Household Travel Survey with 24,806 respondents.

As shown in Fig. 4, l_1 and l_2 are two *consecutive* bus trips of user u. Since u's total walking distance and walking duration are often limited in a day. If u *only walk on feet* during the time between l_1 and l_2, we can assume that a user u's walking scope is bounded in a circle of radius r centered on the alighting bus stop A, and also within a circle of radius r centered on the next boarding stop B. This is because people typically choose to alight at the closest bus stop to her real destination and board at the closest bus stop to her origin (of walking). Note that during the time between A and B, u may either stay at some places (e.g., staying at home during night or working in the office), or walk around (e.g, walking along a shopping street), but still, the walking scope is bounded and the above assumption holds.

By the triangle inequality, it is straightforward to show that the distance between A and B is less than $2r$, as illustrated in Fig. 4. In other words, if a user only travel on feet during an outer-transition, the distance between the alighting stop of the first trip and the boarding stop of the next trip should be less than $2r$, where the number of such possible outer-transition pairs is denoted as k_1. Using the labeled data, we found that the distance of outer-transitions are less than 3.2 km for all the participants. In our experiments, we set $r = 2$km.

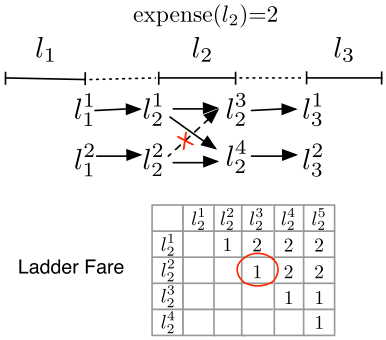


Fig. 5 Reduction of inner-transition using fare constraints

Note that k_1 is possible to be 0 for certain consecutive trips in a segment (obtained through Algorithm 1), which indicates that the user takes other vehicles such as private cars or taxis, instead of walking during the intermediate time between l_1 and l_2 (although the segmentation procedure eliminates non-bus outer-transitions that users pay by smart cards, people still can pay for the taxies by cash). We term such outer-transition as a *drifting point*. In this case, we snip the segment at the drifting point to two pieces and consider each of them as a segment.

In practice, we indexed the bus lines using an R-tree and calculated the k_i pairs for the outer-transition between l_i and l_{i+1} using a range query. After each calculation, we stored the results in a hash table with key (l_i, l_{i+1}), and thus, next time, we can directly retrieve the result from the hash table if we encounter this outer-transition later in the expense records.

- *Fare constraints [for inner-transitions]*

Since we have crawled the fare system (price information) of all the trips that we deal with, we can further reduce the number of inner-transitions by considering the actual expense deducted from the smart card for each trip, after exerting the proximity transition. For example, if the expense calculated by Eq. (1) (using parameters obtained by the method described at Sect. 2.4) for a candidate inner-transition (o_i, d_i) is larger than the actual expense recorded, we prune this candidate. Here, the distance between two bus stops is calculated in advance using the road network distance for all the bus lines (actually, for each stop, we only need to calculate the distance to the first stop of this bus line, and the other mutual distances are easy to obtain) to avoid replicated computation.

Note that the fare constraints not only work for the first and last inner-transition of a segment but also help reduce the inner-transitions in between of two outer-transitions. For example, as illustrated in Fig. 5, suppose (l_1^1, l_2^1), (l_1^2, l_2^2) are 2 candidate outer-transitions from trip l_1 to trip l_2, and (l_2^3, l_3^1), (l_2^4, l_3^2) are 2 candidate outer-transitions from trip l_2 to trip l_3. Before exerting the fare constraints, there are 4 possible inner-transitions in l_2: $l_2^1 \rightarrow l_2^3$, $l_2^1 \rightarrow l_2^4$, $l_2^2 \rightarrow l_2^3$, and $l_2^2 \rightarrow l_2^4$; however, by checking the price information shown in the ladder-fare table, we find that the fare from l_2 to l_3 is less than the actual expense recorded. Therefore, $l_2^2 \rightarrow l_2^3$ is not a possible inner-transition for l_2, and thus, the total number of candidate transitions from the alighting stop of l_1 to the boarding stop of l_3 is decreased from 4 to 3.

- *Temporal constraints [for inner and outer transitions]*

Although we have only one time stamp for a trip, we can leverage it to further weed out unreasonable candidates. Let t_1, t_2, t_3 be the time stamps of three trips $l_i = (o_i, d_i)$, for $i = 1, 2, 3$. Recall that for ladder-fare trips, the time stamps are the alighting time, and

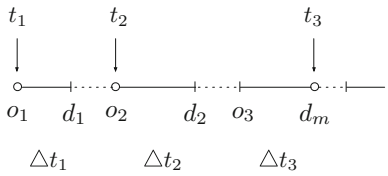


Fig. 6 Temporal constraints

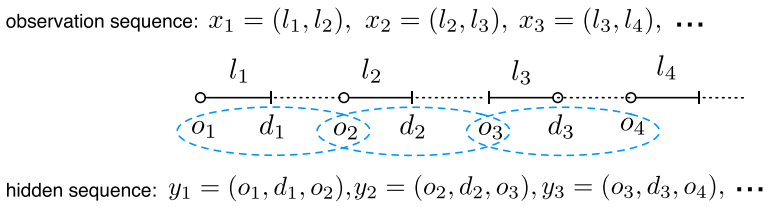


Fig. 7 Constructed observation sequence and hidden sequence in the linear-chain CRF

for non-ladder-fare trips, the time stamps are boarding time. As shown in Fig. 6, l_1 and l_2 are non-ladder-fare trips, and l_3 is a ladder-fare trip. The minimum travel time between o_i and d_i, denoted as $\triangle t_i$, can be calculated using the road network, where the travel speed is substituted with the limit driving speed (refer to Sect. 2.3). Consequently, the following conditions should hold for this example:

$$\triangle t_1 \leq t_2 - t_1, \tag{5}$$

$$\triangle t_2 + \triangle t_3 \leq t_3 - t_2. \tag{6}$$

$$\cdots$$

Thus the candidates which violate the above conditions should be removed.

To avoid duplicate calculations, we also pre-compute all the minimum travel time between a given bus stop to the 0-coded bus stop of each bus line (thus we have the minimum travel time between any pair of bus stops) and store the results using a hash table.

Next, we introduce a unified model to deal with all the above constraints, and incorporate the labeled trips (note that until now we do not rely on any labeled data) to infer the most possible candidate for a segment.

3.3 Semi-supervised CRF with constraints

3.3.1 Model

Actually, given the candidates generated for a segment, our problem can be formulated as a sequential labeling problem. Conditional Random Fields (CRFs) [15] have been successfully applied to many sequential labeling applications in data mining and machine learning. Specifically, we construct a linear-chain CRF as follows. Given a segment $S = \{l_1, l_2, \ldots, l_m\}$, let $\mathbf{x} = \{x_1, x_2, \ldots, x_{m-1}\}$ be the observation sequence, where $x_i = (l_i, l_{i+1})$ for $i = 1, 2, \ldots, m-1$. That is, the outer-transition between consecutive lines is regarded as a node in the CRF chain (note that here each node is a pair of trips, as shown in Fig. 7). Later, let $y_i = (y_i^1, y_i^2, y_i^3)$ denote the triple (o_i, d_i, o_{i+1}) for $i = 1, 2, \ldots, m-1$, which is an inner-transition coalesced with an outer-transition (we will crystallize the reason for this later). The sequence $\mathbf{y} = \{y_1, y_2, \ldots, y_{m-1}\}$ is thus the label sequence.

With fully labeled sequences, CRF is typically trained by maximizing the penalized conditional log-likelihood on the training sequences $\mathcal{D}$ with length N

$$L(\lambda, \mathcal{D}) = \sum_{i=1}^{N} \log p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) - \frac{\sum_k \lambda_k^2}{2\sigma^2}, \tag{7}$$

$$p_\lambda(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\lambda)} \exp\left(\sum_{i=1}^{m-1}\sum_{k=1}^{K} \lambda_k f_k(y_i, y_{i+1}, \mathbf{x})\right), \tag{8}$$

$$Z(\lambda) = \sum_{\mathbf{y}} \exp\left(\sum_{i=1}^{m-1}\sum_{k=1}^{K} \lambda_k f_k(y_i, y_{i+1}, \mathbf{x})\right). \tag{9}$$

Here, to avoid over-fitting, we include a Gaussian prior with zero mean and variance $\sigma^2 = 10$.

However, in our dataset, a significant number of segments are partially labeled (for certain bus lines), a semi-supervised training approach that can take the full use of available labels is more preferred. More importantly, rich prior knowledge and constraints we derived are not thoroughly leveraged. Therefore, we employ the generalized expectation criterion [20] as an objective function, which enables semi-supervised CRF training with constraints as side information. Given a real-valued constraint function $G(\mathbf{y}, \mathbf{x})$ and unlabeled data $\mathcal{U}$, the Generalized Expectation Criterion is given by

$$O(\lambda, \mathcal{D}, \mathcal{U}) = L(\lambda, \mathcal{D}) - S(E_{\tilde{p}(\mathbf{x})}\left[E_{p_\lambda(\mathbf{y}|\mathbf{x})}\left[G(\mathbf{y}, \mathbf{x})\right]\right]), \tag{10}$$

where $\tilde{p}(\mathbf{x})$ is the empirical distribution over unlabeled data $\mathcal{U}$, $E[\cdot]$ stands for the expectation, and S is a score function[7] expressing the distance between the model expectation and a targeted expectation. The optimization of Eq. (10) can be performed using gradient-based methods [9].

3.3.2 Collaborative filtering-based feature engineering

Compared with other sequential models such as Hidden Markov Model (HMM), CRF is more flexible in incorporating features, e.g., the transition from one label to another can also depend on the whole observation sequence. As such, designing features is the most critical part for applying CRF to various applications. We first extract typical uni-gram and bi-gram features for the CRF model as follows.

Segment features We use both uni-gram features and bi-gram for segment features. For each uni-gram label y_i, the features we used for training include:

- the indicator function of l_i and x_i;
- the trip type (non-ladder or ladder);
- time interval $\triangle t = t_{i+1} - t_i$, which is discretized to hours;
- expense of trip l_i, which is integer multiples of the unit price.

The bi-gram features include:

- the indicator functions of (y_i, y_{i+1}), (x_i, y_{i+1}), and (x_i, y_i, y_{i+1});
- time interval $\triangle t_{i+2} - t_i$, which is discretized to hours;
- whether $y_i^3 = y_{i+1}^1$;
- whether l_i and l_{i+2} are the same bus line.

Table 3 Overall statistics on number of triples versus number of distinct triples in labeled segments

	Number of visited triples	Number of visited distinct triples
User	18.6	7.4
Time slots	21,371	9,548

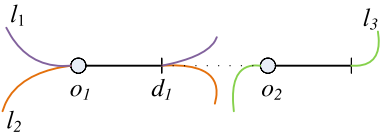


Fig. 8 An example of two bus lines share overlapping bus stops

Furthermore, to adequately exploit the labeled data and reduce data sparsity, we employ collaborative filtering approaches to derive latent features, in addition to the typical uni-gram/bi-gram features for each sequence (i.e., segment). The latent features are categorized into two types: User Interest Features (UI) and Temporal Interest Features (TI).[8]

Specifically, on one hand, user's mobility patterns are typically regular [25], e.g., an office worker would probably choose the identical boarding stop and alighting stop every day commuting between home and work places. Thus, the historical frequency is a strong signal for identifying whether a triple label $y = (y^1, y^2, y^3)$ is the actual choice. Table 3 shows that on average, the number of distinct triples is much fewer than the number of triples, which verifies that only a small number of triples are frequently visited by a user. Therefore, if we observe that a triple has occurred in a user's historical records, it is more likely to be the actual choice compared to the unprecedented triples. Note that since many bus lines share overlapping bus stops in a certain area, we consider the triples that belong to different bus lines as the same triple. For example, as shown in Fig. 8, $o_1 \rightarrow d_1$ is shared by two bus lines, and we treat $y = (o_1, d_1, o_2)$ as a single triple label instead of two candidates. On the other hand, the empirical frequency itself is not able to fully demonstrate a user's interest, or rather, an undiscovered triple might also have been visited by the user, but fails to be recorded in the historical data because that the bus line is a non-ladder-fare line. For example, assume that in Fig. 8, l_1 and l_3 are ladder-fare lines, while l_2 is a non-ladder-fare line, a user u_1 frequently choose (o_1, d_1, o_2) through the transition $l_1 \rightarrow l_3$, while another user u_2 frequently choose (o_1, d_1, o_2) through $l_2 \rightarrow l_3$. However, u_1's records will be fully recorded, while the boarding/alighting stops (o_1 and d_1) of u_2's transition $l_2 \rightarrow l_3$ are totally ignored since l_2 is a non-ladder-fare line. In view of this, we apply collaborative filtering, which is widely adopted in recommender system community, to solve the data sparsity issue with the "crowd knowledge" exploited from all the labeled trips. Imagine that if u_1 and u_2 behave similarly in a way reflected in the historical records (e.g., they live/work at places close to each other and frequently choose the same stops for communicating), collaborative filtering methods will identify u_2's preference for (o_1, d_1, o_2) in line with u_1. Likewise, Table 3 also shows that in terms of temporal space (the partition of time slots will be detailed later), some adopted triple labels would repeat previous patterns, and thus, we also apply collaborative filtering to extract temporal interest over the triples. The procedure of engineering the two latent features is detailed as follows.

[7] We employed the square distance as the score function in our implementation, following [9].

[8] We adopt the term "interest" following the idioms used in the recommender system community.

User interest feature (UI) UI measures how likely a user u would choose a particular triple label $y = (y^1, y^2, y^3)$, which reflects this user's general interest on different possible triples. The triple set contains all possible transitions of all bus lines by filtering by previous-derived constraints (notice that the filtering is user-centric instead of segment-centric; therefore, only proximity constraints take effect). As mentioned earlier, all triples with completely identical bus stops are also merged as a single one. The frequency of user u at triple y is represented as $c(u, y)$, which is the element of user-triple matrix. Next, we apply matrix factorization to uncover users' hidden interests for triples. Given user u and triple y, the interest score is denoted by

$$f_{uy}^{UI} = \langle p_u, q_y \rangle, \tag{11}$$

where $q_y \in \mathbb{R}^{K_{UI}}$ and $p_u \in \mathbb{R}^{K_{UI}}$ are latent vectors which describe the interest. All users' transition patterns are reduced and represented in the low dimensional latent space. In this case, users with similar patterns in the latent space can fill the missing values of each other at the potentially interesting but previous unrecorded triples ($c(u, y) = 0$), and thus, the sparsity problem can be effectively alleviated.

Temporal interest feature (TI) User mobility patterns are discrepant from time to time, e.g., users would usually commute between home and work places on weekdays; however, on weekends or holidays, they probably go for a tour and the mobility pattern would be obviously different from that of weekdays. Even in different time periods of a day, in the morning, the triples frequently adopted are those from home to work places, while in the evening, the adopted triples are those in the adverse direction. Considering the typical commuting patterns in the studied city, we discretize the time of a day to the following time slots 12 a.m.–6 a.m., 6 a.m.–9 a.m., 9 a.m.–12 p.m., 12 p.m.–5 p.m., 5 p.m.–8 p.m., 8 p.m.–12 a.m. and set apart different days of a week (that is, we have in total $6 \times 7 = 42$ time slots). Similar to the generation of UI feature, we apply matrix factorization to automatically discover the hidden temporal-triple interest implicated in the original temporal-triple matrix (notice the triple set is the same as that in UI feature generation step). Given the discretized time t and triple y, the interest score is represented by

$$f_{ty}^{TI} = \langle w_t, h_y \rangle, \tag{12}$$

Here, we learn the four latent vectors in the UI and TI generation procedure by means of Bayesian-personalized ranking matrix factorization (BPRMF) [23] on the training-labeled trips. The reason we adopt BPRMF instead of other matrix factorization methods is that it models personalized pairwise preference between visited triples and unvisited triples, the strategy of directly optimizing ranking criteria keeps line with our ultimate goal (rank interested triples higher than others). Compared to the original BPRMF, the only difference in our scenario is that we sample the unvisited triple j from the neighborhood of the actually visited triple i, while the original algorithm would sample from the whole unvisited triples, in other words, it makes more sense to only compare pairwise preference between triples that are not far away instead of between all the triples.

3.3.3 Constraints

In our model, the general expectation criterion makes very natural an under-explored paradigm for incorporating an abundant amount of prior knowledge. Suppose, we would like to build a system to extract information from citations of research papers such as *authors, title, journal*, we actually know that the word *ACM* should usually be part of a *journal*. If given one distribution where *ACM* is usually part of a *journal* and another where *ACM* is

usually a part of a *title*, we know that the first distribution is preferable. That is, the prior knowledge tells us desirable properties of the distribution over output variables. The distribution we want to match deriving from the prior knowledge is called target distribution, e.g., we expect that $p(journal|ACM) = 0.99$, $p(title|ACM) = 0.01$. By designing constraint function $\phi(\mathbf{x}, \mathbf{y})$, the CRF model will also generate a distribution over the constraint function as follows

$$E_\lambda[\phi] = E_{p_\lambda(\mathbf{y}|\mathbf{x})}[\phi(\mathbf{x}, \mathbf{y})] = \sum_{\mathbf{y}} p_\lambda(\mathbf{y}|\mathbf{x})\phi(\mathbf{x}, \mathbf{y}) \tag{13}$$

The general expectation criterion encourages the gap (e.g, the square distance applied in this article) between the target distribution and model expectation distribution $E_\lambda[\phi]$ is as small as possible. For the constraint functions, them should be designed as

$$\phi_{journal,ACM}(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{n} 1_{\{y_j=journal\}} 1_{\{x_j=ACM\}} \tag{14}$$

$$\phi_{title,ACM}(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{n} 1_{\{y_j=title\}} 1_{\{x_j=ACM\}} \tag{15}$$

where the indicator function 1_P returns 1 if the predicate P is true, and 0 otherwise.

In our scenario, the constraints are categorized into two types: one-label constraints and two-label constraints. One-label constraints, which restrict the candidates of a single label y_i, are associated with the features including both inner-transitions and outer-transitions, such as proximity constraints, fare constraints and temporal constraints with the form of Eq. (5) (termed as Type I). Two-label constraints, which restrict the conditional probability of two consecutive candidates, are associated with the features including the temporal constraint with the form of Eq. (6) (termed as Type II), and

$$y_i^3 = y_{i+1}^1, \forall i = 1, 2 \ldots, m-1, \tag{16}$$

where $y_i = (y_i^1, y_i^2, y_i^3)$. This is to ensure the chain is connected as shown in Fig. 7.

To obtain the target distribution for both types of constraints, we follow the method described in [19], which makes use of the majority labeled features. The detail of this approach is like this: for a given feature, we first find out the majority label in the limited labeled instances and then use a simple heuristic to derive distributions from majority label information: we assign 0.99 probability to the majority label and divide the remaining probability uniformly among the remainder of the labels.

Both one-label and two-label constraints can be represented by constraint functions $G(\mathbf{y}, \mathbf{x})$ as follows:

– proximity, fare, Type I temporal constraint (one-label). Since their formations are similar, we only take proximity constraint function for example. Outer-transitions with a distance smaller than $2r$ are assigned higher probability by the constraint function

$$G_r(\mathbf{y}, \mathbf{x}) = \sum_{i=1}^{m} \phi_r(\mathbf{x}, y_i), \tag{17}$$

where the *constraint feature*

$$\phi_r(\mathbf{x}, y_i) = \begin{cases} 0.99/a_i & distance(y_i^2, y_i^3) < 2r \\ 0.01/b_i & distance(y_i^2, y_i^3) \geq 2r \end{cases} \tag{18}$$

assuming $\{C_i\}_{i=1}^{c_i}$ is a set of all candidate states of y_i, $a_i = |\{C_i | distance(C_i^2, C_i^3) < 2r\}|$, $b_i = c_i - a_i$. Note that we given probability $0.01/b_i$ to these impossible candidates instead of 0, because it will lead to a smoother solution.

– Type II temporal constraint (two-label). Outer-transitions with a time spent larger than the minimal travel time are assigned higher probability by the constraint function

$$G_{II}(\mathbf{y}, \mathbf{x}) = \sum_{i=1}^{m-1} \phi_{II}(\mathbf{x}, y_i), \tag{19}$$

where the *constraint feature*

$$\phi_{II}(\mathbf{x}, y_i) = \begin{cases} 0.99/d_i & t_{y_{i+1}^2} - t_{y_i^1} >= minimal\ travel\ time \\ 0.01/e_i & others \end{cases} \tag{20}$$

$d_i = |\{(C_i, C_{i+1}) | t_{C_{i+1}^2} - t_{C_i^1} >= minimal\ travel\ time\}|$, $e_i = c_i \times c_{i+1} - d_i$.

As a result, all the constraints we derived before are incorporated into this framework succinctly and consistently, which is the reason that we model a triple as a node in a linear-chain CRF. Regarding each boarding or alighting stop of a bus trip as a hidden state could yet be an alternative way to model a segment, which forms a high-order CRF; however, additional computation cost is exponential to the order of the CRF [24] (in our scenario, the order should be 3 due to the distinct properties of inner and outer transitions). On the contrary, by connecting the inner and outer transitions with a triple, we can naturally restrict the candidates to a relatively small set and thus considerably accelerate the inference. In addition, our model does not need to separately tackle different higher-order constraints and features with various forms, which might clutter the model, yet some intrinsic prior knowledge and strict conditions, such as fare constraints between inner-transitions, are naturally leveraged to pre-exclude irrelevant labels and redundant features.

4 Evaluation

4.1 Settings

The dataset we used is described in Sect. 2. Here, we introduce (1), which baseline methods were compared with, and (2) how we evaluated these methods.

4.1.1 Baselines

We compared our method (semi-supervised **CRF** with **S**egment features, **U**ser interest features, and **T**emporal features, optimized with **C**onstraints), shortened as "CRF(SUT)+C" against the following baselines.

– CRF with only segment features and constraints ("CRF(S)+C" for short). This algorithm uses the same settings as CRF(SUT)+C, except that it does not incorporate the collaborative user interest and temporal interest features. This is for evaluating whether the collaborative features are useful to detect the bus stops in a trip.
– CRF with segment features, collaborative features learned by basic matrix factorization and constraints ("CRF(BM)+C" for short). This algorithm uses the same settings as CRF(SUT)+C, except that user interest features and temporal interest features are learned

by basic matrix factorization instead of BPRMF. This is for evaluating whether BPRMF algorithm outperforms basic factorization method, which directly optimizes mean square loss.
- CRF without constraints ("CRF(SUT)" for short). This algorithm uses the same setting as CRF(SUT)+C, except that it does not incorporate constraints. This is for evaluating whether the constraints are useful to detect the bus stops in a trip.
- Trip-Chaining with maximum frequency ("TC+MF" for short). The Trip-Chaining (TC) algorithm is adopted by most existing approaches [2,7,29,31] for inferring origin–destination pairs. TC is based on several explicit assumptions such as the proximity between consecutive trips and "the first trip of a day starts from the alighting station of last night" [7]. Since TC requires at least one stop to be known for all trips, in case TC fails to find the stop of some trip in a segment, we assigned it with the most frequent label in the labeled test data.
- Trip-Chaining with maximum similarity ("TC+MS" for short), which is a state-of-the-art variation of the trip-chaining method [26]. A major difference between TC+MS and TC is that TC+MS assigns similar destinations (origins) to trips that have similar origins (destinations) when other rules in TC fail.

4.1.2 Criteria

We measured the performance of each algorithm using accuracy, calculated by

$$\text{Accuracy} = \frac{\text{correctly identified unlabeled bus stops}}{\text{unlabeled bus stops}}. \tag{21}$$

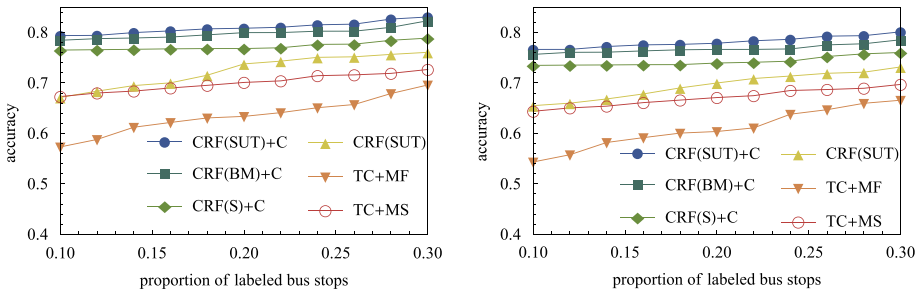
For each individual, we calculate the accuracy after running a test for each method. The overall accuracy is calculated by an average of tenfold cross-validation.

4.2 Evaluation on all card holders' data

We first evaluated our method on all card holders' data using labeled trips as the testing set (otherwise, the ground truth is not known). Specifically, we first selected fully labeled segments (of which we removed 8.5 % segments with length less than 3) after performing Algorithm 1. In order to reveal the performance of these methods on both labeled and unlabeled trips, we randomly removed labels for 70–90 % trips, which resulted in the remaining 10–30 % labeled trips to fit the same scale of labels as the whole dataset (note that in the entire dataset 26.54 % trips are labeled). Next, we further randomly removed a bus stop (either boarding or alighting) for each trip, so as to compare our methods against TC-based approaches (where they require at least one bus stop is known for each trip). Then, we conducted tenfold cross-validation to calculate the accuracy of each method.

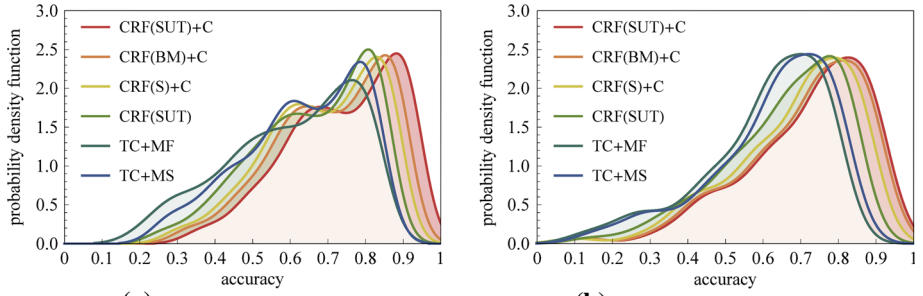
Figure 9a plots the overall accuracy, where the x-axis is the proportion of labeled bus stops. It is clear that our method significantly outperforms the competitors. For example, even with only 10 % labeled bus stops, CRF(SUT)+C achieves a high accuracy at 0.8, while the performance of other methods, especially the TC+MF method, rely much more on the labeled data. Comparing CRF(SUT)+C with CRF(S)+C, it shows that collaborative features have improved the result evidently. We also see that CRF(SUT)+C outperforms CRF(BM)+C, which implies the efficiency of BPRMF algorithm. Note that the result of TC+MS is in good agreement with the reported (66 %) performance mentioned in [26].

Next, we investigated the distribution of accuracy among all users. Figs. 10a and 11a, respectively show the probability distribution function (PDF) (fitted from the histogram)



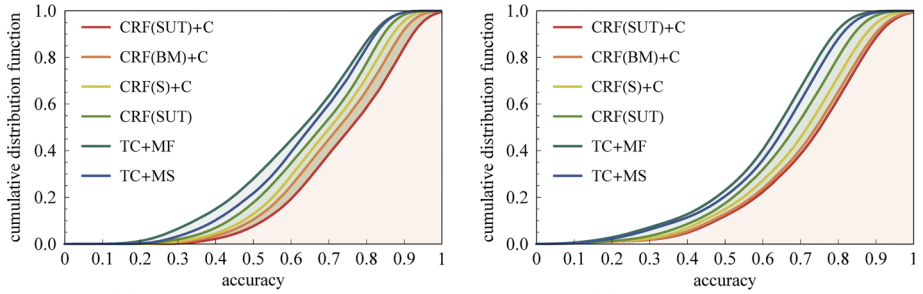
(a) overall accuracy of all users' results **(b)** overall accuracy of participants' results

Fig. 9 Overall accuracy of all users' results versus participants' results. **a** Overall accuracy of all users' results, **b** overall accuracy of participants' results



(a) PDF of all users' results **(b)** PDF of participants' results

Fig. 10 PDF of all users' results versus participants' results. **a** PDF of all users' results, **b** PDF of participants' results



(a) CDF of all users' results **(b)** CDF of participants' results

Fig. 11 CDF of all users' results versus participants' results. **a** CDF of all users' results, **b** CDF of participants' results

and the cumulative distribution function (CDF) of the accuracy among all users. The results validate the advantage of our method, e.g., Fig. 10a shows that the accuracy of our method still has a high probability density at the position of 0.9.

4.3 Evaluation on completely labeled participants' data

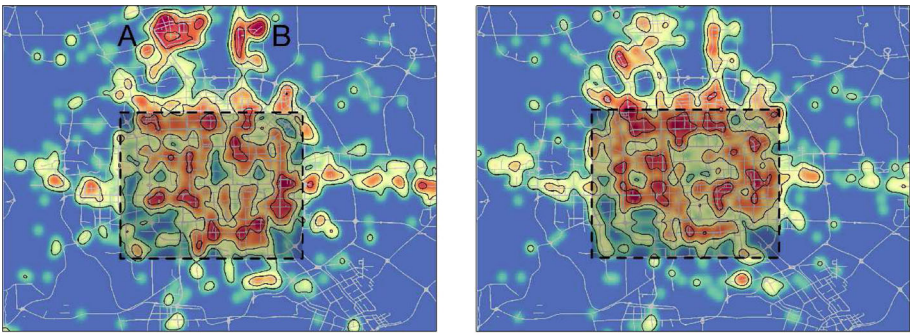
As mentioned in Sect. 2.4, the 102 participants manually labeled all their bus trips, including non-ladder-fare and ladder-fare trips. Basic demographic information of their age and gender

Table 4 Demographics of the participants

Gender		Age			
Male	Female	19–24	25–30	31–36	37–47
57.6 %	42.4 %	39.4 %	45.5 %	10.6 %	4.5 %

Table 5 The running time (minute) of all methods

	CRF(SUT)+C	CRF(BM)+C	CRF(S)+C	CRF(SUM)	TC+MF	TC+MS
All users	34.71	41.15	28.98	35.23	50.91	52.27
Participants	1.26	1.41	1.17	1.32	1.42	1.53



(a) home **(b)** work

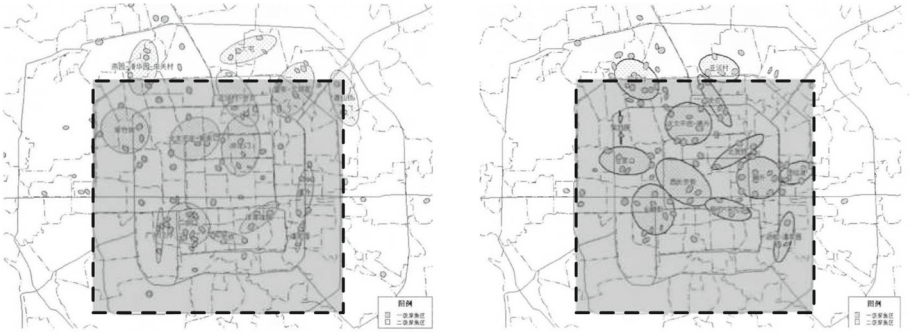
Fig. 12 Identified distribution of home and working places of all card users using 2D kernel density estimation. **a** Home, **b** work

is presented in Table 4. Thus, we have both non-ladder-fare trips and ladder-fare trips in the testing set, which is exactly the situation in the real data. Similarly, as the above experiment, we constantly fed 10–30 % (randomly chosen) trips with the boarding or alighting stops as labels, then compare all methods using tenfold cross-validation.

As shown in Figs. 9b, 10b and 11b, the accuracy, PDF and CDF exhibit consistent trend with the previous results of all card holders' data. We found that the accuracy for all the methods are actually a little lower than the previous experiment; however, our method still shows clear advantage compared with other methods. In particular, the overall accuracy of our method is still close to 0.8 when we have 25 % labeled stops. Furthermore, the running time of all methods for both all users and participants are listed in Table 5.

4.4 Detection of important places

As a demonstration, we show how the enriched smart card transactions can be utilized to mine important locations such as home and work places of users. As shown in Fig. 9a, the overall accuracy of our method is higher than 0.8 given the 25 % labeled stops. Hence, we applied the proposed method to the entire dataset, and reconstructed mobility history for each individual. We employed the clustering-based method proposed in [14] to identify home and work places. Later, we performed a 2D Kernel Density Estimation (KDE) given the detected



(a) home **(b)** work

Fig. 13 Distribution of home and working places of the studied city in 2009 household survey. **a** Home, **b** work

home and work places, as shown in Fig. 12a and b, respectively. Furthermore, according to the household survey in 2009 [4], the distribution of home as well as work places of the studied city is shown in Fig. 13a, b. Comparing Fig. 12a with 13a and Fig. 12b with 13b, the *shaded areas* imply that the identified hot spots for both home and work places coincide well with the local household surveys. Figure 12a also shows two hot spots outside the *shaded area*, where region A corresponds to Huilongguan and region B corresponds to Tiantongyuan, both of which are dense residential zones developed in recent years (actually Huilongguan has become the largest independent living community in Asia.[9]) For the 102 participants, we compared the identified home and work places with the real ones provided by themselves, where we successfully identified 97 home places and 95 work places.[10] However, if we directly use the smart card data without applying our space alignment approach, only 53 home places and 69 work places are successfully identified, i.e., the space alignment approach increases the performance by 83 % for home identification and 38 % for work place identification. The reason behind is that many bus stops that are close to the real home or work places are actually derived from the non-ladder-fare trips, which are implicitly learned by the proposed method.

5 Related work

5.1 Human mobility analytics

Human mobility is attracting more researchers' attentions thanks to the increasing availability of mobility data. Recent studies converge to suggest that human mobility is highly regular, predictable, and unique. Ground-breaking work on studying human mobility patterns from large-scale mobile phone traces is established in [12]. In a series of work, they reported that human mobility patterns show high degree of spatial and temporal regularity, and each individual has a significant probability to return to a few highly frequented locations. Based on a 3-month mobile records captured from 50,000 individuals, Song et al. [25] suggested that human mobility has a predictability of 93 %. More recently, de Montjoye et al. [21]

[9] http://baike.baidu.com/view/1318898.htm.

[10] According to our privacy agreement with the participants, we cannot show the density distribution of their home and work places (as Fig. 12) here.

mathematically formulated the uniqueness of mobility and showed that four spatial–temporal points are enough to uniquely identify 95 % of the individuals, based on an investigation of a 15 months mobility data covering 1.5 million individuals.

Mining human mobility data has also enabled a variety of emerging applications. For example, Yuan et al. [30] introduced a driving direction system with the intelligence mined from local taxi drivers. Ge et al. [10] presented a recommendation system in order to maximize the profit of a taxi driver, based on taxi trajectories. Hoh et al. [13] designed a time-to-confusion metric and a cloaking algorithm to help users avoid privacy risks based on vehicle GPS trajectories. Meanwhile, mobility data have been utilized for studying several research topics in social science such as friendships and social ties [28]. For example, Cranshaw et al. [6] showed that human mobility patterns have strong connections with the structure of their underlying social network.

In this paper, motived by the above work, we restrict ourselves to the problem of recovering human mobility data from transaction data associated with bus trips. Compared with other kinds of human mobility data such as mobile phone traces or check-ins, public transit data characteristically reveal individual's daily transits between important locations such as home and work places, which may complement existing approaches and findings founded on other types of mobility data. Additionally, the methods provided in this paper might help identify new opportunities in human mobility analytics dealing with cross-application data.

5.2 Mining smart card transactions

Smart cards and integrated ticketing are supported by public transit operators in many cities, which provides convenience to both citizens and governments for public transit ticketing. The overwhelming usage of smart card makes the transaction data invaluable resources for understanding urban commute patterns and human dynamics.

In transportation research area, numerous studies have attempted to mine users' travel behaviors from smart card transactions [1], For example, Utsunomiya et al. [27] reported several findings on walking access distance, frequency and consistency of daily travel patterns, and variability of smart card customer behaviors by residential area, based on smart card transaction data in combine with card holders' personal information, and proposed to improve user trust in transit service, and adjust fare according to users' needs. Recently, Ceapa et al. [5] investigated the crowdedness of London Underground by mining the spatial–temporal patterns from the Oyster Card Data. Their results indicate that the crowdedness is highly regular and predicable and suggest users slightly adjust their travel time to avoid congestion peak. Pelletier et al. [22] provided a comprehensive review of recent literatures on the usage of smart card data in public transit planning.

Existing work on mining smart card transactions, especially bus trip transactions, often encounters the problem of data incompleteness. This is because most Automatic Fare Collection (AFC) systems record the bus trip boarding location coarsely at the bus-route level (without the information of specific boarding and alighting stops). For example, as mentioned in [16] and [29], the London Oyster Card data only contain the information of origin and start time for bus trips, since the pricing of a bus trip does not depend on the destination (but for rail/tube trips, the destination is also recorded). Similarly, in the dataset used by Liu et al. [18] to mine collective mobility patterns, the bus trips only have information of boarding time and travel fare.

Several approaches have been proposed to infer the boarding stops [7] or alighting stops [26] of a bus trip. Nevertheless, these approaches still require that at least one location (either the boarding or alighting stop) is available. For example, Trépanier et al. [26]

addressed the problem of inferring trip destinations with smart card transactions where the boarding stop is recorded. Most of these approaches employ the Trip-Chaining method or its variations [2,7,26,29,31], which is based on several assumptions, such as the users return to the first boarding station at the end of a day [26]. Cui [7] suggested that side information such as the Automatic Vehicle Location (AVL) data could be leveraged to infer the origin and destinations when location information is not available in the smart card transactions.

Our work is different from the above methods in the following aspects: First, we provided a space alignment framework to coalesce the information in the monetary space (rarely considered before), temporal space (with a historical view instead of separated days adopted by many existing approaches), and geospatial space, which is flexible enough to be applied to different datasets with various types of missing information, e.g., even for trips that neither alighting nor boarding stops are available, our approach can still infer the origins and destinations with a high accuracy (75 %), which improves the state of the art with 10 % when labels are rare (<25 %). Second, instead of using hard-coded inference rules and assumptions, we employed a probabilistic model, which naturally incorporates domain constraints and inherits the advantage of statistical modeling to achieve a global optimization. Finally, due to the lack of large-scale ground truth data for testing the accuracy of a model, few existing approaches have evaluated the rate of correctly inferred bus stops. In contrast, we directly validated the proposed method using a large-scale human-labeled data, where every trip that appeared in the transactions during the 4 months is labeled by the participant herself.

Nevertheless, our method is motivated by existing approaches, and we believe the proposed method as well as the reconstructed data would be beneficial for urban planners, transportation engineers, and researchers in related fields.

6 Conclusion

We have provided a systematic way to recover individual mobility history from urban-scale smart card transactions. By aligning data in different dimensions, we formulated several underlying constraints from the transaction data, and incorporated these constraints into a semi-supervised probabilistic model with latent features derived using collaborative filtering. Extensive experiments validated that the proposed method has a considerably high accuracy given very limited number of known alighting or boarding stops.

Although the work reported in this paper is based on a public transit transaction dataset, we believe the proposed space alignment framework can be easily adapted to other location-related transaction data and may also provide implications to data miners who deal with cross-application datasets.

References

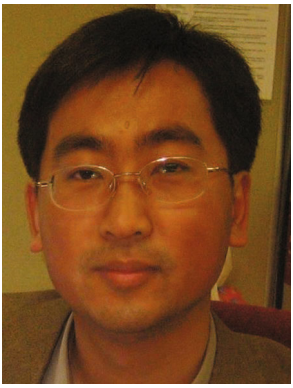
1. Agard B, Morency C, Trépanier M (2006) Mining public transport user behaviour from smart card data. In: 12th IFAC symposium on information control problems in manufacturing-INCOM, pp 17–19
2. Barry JJ, Freimer R, Slavin H (2009) Use of entry-only automatic fare collection data to estimate linked transit trips in New York city. Transp Res Rec J Transp Res Board 2112(1):53–61
3. Bassett DR Jr, Wyatt HR, Thompson H, Peters JC, Hill JO (2010) Pedometer-measured physical activity and health behaviors in United States adults. Med Sci Sports Exerc 42(10):1819
4. Bin M (2009) The spatial organization of the separation between jobs and residential locations in beijing. Acta Geogr Sinica 12:009
5. Ceapa I, Smith C, Capra L (2012) Avoiding the crowds: understanding tube station congestion patterns from trip data. In: Proceedings of the ACM SIGKDD international workshop on urban computing, pp 134–141

6. Cranshaw J, Toch E, Hong J, Kittur A, Sadeh N (2010) Bridging the gap between physical location and online social networks. In: Ubicomp, pp 119–128
7. Cui A (2006) Bus passenger origin-destination matrix estimation using automated data collection systems. Master's thesis, Massachusetts Institute of Technology
8. Daniels R, Mulley C (2011) Explaining walking distance to public transport: the dominance of public transport supply. World 28:30
9. Druck G, Mann G, McCallum A (2009) Semi-supervised learning of dependency parsers using generalized expectation criteria. In: Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP, vol 1, pp 360–368
10. Ge Y, Xiong H, Tuzhilin A, Xiao K, Gruteser M, Pazzani M (2010) An energy-efficient mobile recommender system. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 899–908
11. Ginsberg J, Mohebbi MH, Patel RS, Brammer L, Smolinski MS, Brilliant L (2008) Detecting influenza epidemics using search engine query data. Nature 457(7232):1012–1014
12. Gonzalez MC, Hidalgo CA, Barabasi AL (2008) Understanding individual human mobility patterns. Nature 453(7196):779–782
13. Hoh B, Gruteser M, Xiong H, Alrabady A (2010) Achieving guaranteed anonymity in GPS traces via uncertainty-aware path cloaking. IEEE Trans Mob Comput 9(8):1089–1107
14. Isaacman S, Becker R, Cáceres R, Kobourov S, Martonosi M, Rowland J, Varshavsky A (2011) Identifying important places in peoples lives from cellular network data. In: Pervasive computing, pp 133–151
15. Lafferty JD, McCallum A, Pereira FCN (2001) Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the eighteenth international conference on machine learning, ICML '01, pp 282–289
16. Lathia N, Capra L (2011) Mining mobility data to minimise travellers' spending on public transport. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 1181–1189
17. Lathia N, Froehlich J, Capra L (2010) Mining public transport usage for personalised intelligent transport systems. In: 2010 IEEE 10th international conference on data mining (ICDM), IEEE, pp 887–892
18. Liu L, Hou A, Biderman A, Ratti C, Chen J (2009) Understanding individual and collective mobility patterns from smart card records: a case study in shenzhen. In: Intelligent transportation systems, 2009. ITSC'09, IEEE, pp 1–6
19. Mann G, McCallum A (2008) Generalized expectation criteria for semi-supervised learning of conditional random fields. In: Proceedings of ACL, pp 870–878
20. Mann GS, McCallum A (2010) Generalized expectation criteria for semi-supervised learning with weakly labeled data. J Mach Learn Res 11:955–984
21. de Montjoye YA, Hidalgo CA, Verleysen M, Blondel VD (2013) Unique in the crowd: the privacy bounds of human mobility. Scientific Reports 3
22. Pelletier MP, Trépanier M, Morency C (2011) Smart card data use in public transit: a literature review. Transp Res Part C Emerg Technol 19(4):557–568
23. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence, AUAI Press, pp 452–461
24. Sarawagi S, Cohen WW (2004) Semi-markov conditional random fields for information extraction. Adv Neural Inf Process Syst 17:1185–1192
25. Song C, Qu Z, Blumm N, Barabási AL (2010) Limits of predictability in human mobility. Science 327(5968):1018–1021
26. Trépanier M, Tranchant N, Chapleau R (2007) Individual trip destination estimation in a transit smart card automated fare collection system. J Intell Transp Syst 11(1):1–14
27. Utsunomiya M, Attanucci J, Wilson N (2006) Potential uses of transit smart card registration and transaction data to improve transit planning. Transp Res Rec J Transp Res Board 1971(1):119–126
28. Wang D, Pedreschi D, Song C, Giannotti F, Barabasi AL (2011a) Human mobility, social ties, and link prediction. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1100–1108
29. Wang W, Attanucci JP, Wilson NH (2011b) Bus passenger origin-destination estimation and related analyses using automated data collection systems. J Public Transp 14(4)
30. Yuan J, Zheng Y, Xie X, Sun G (2013) T-drive: enhancing driving directions with taxi drivers' intelligence. IEEE Trans Knowl Data Eng 25(1):220–232
31. Zhao J, Rahbee A, Wilson NH (2007) Estimating a rail passenger trip origin-destination matrix using automatic data collection systems. Comput Aid Civil Infrastruct Eng 22(5):376–387

Fuzheng Zhang is currently a Ph.D. Candidate supervised jointly by University of Science and Technology of China and Microsoft Research Asia. His research Mainlinie focuses on consumption and location-based user understanding by using techniques such as machine learning, data mining, big data analysis, etc. He has authored many top-tier international conference papers and journal articles in his research area, such as WWW, Ubicomp, TIST. He has received the best paper award in ICDM2013. He is currently working as a fulltime intern in the Social and Urban Mining (SUM) Group of Microsoft Research Asia. He received his B.S. degrees both in Computer Science and Statistics and Finance from the University of Science and Technology of China in 2010. His research interests include behavior data mining, spatial temporal data mining, ubiquitous computing, recommender system and machine learning.

Nicholas Jing Yuan is an associate researcher in the Social and Urban Mining (SUM) Group of Microsoft Research Asia. He is also a member of the Ubiquitous Computing Group (Ubicomp) in Microsoft Research. He got his Ph.D. degree in Computer Science from the School of Computer Science and Technology in 2012, and he received his B.S. degree in Mathematics from the School of the Gifted Young in 2007, both in University of Science and Technology of China. From 2009 to 2012, he worked as a full time research intern in MSR Asia. Currently, his research interests include behavioral data mining, spatial-temporal data mining and computational social science. He is a Microsoft Fellow, a member of CCF (China Computer Federation), ACM, and IEEE.

Yingzi Wang is currently a Ph.D. Candidate supervised jointly by University of Science and Technology of China and Microsoft Research Asia. She recevied her B.S. degree in computer science from the University of Science and Techonology of China in 2013. She worked as a fulltime intern in the Social and Urban Mining (SUM) Group of Microsoft Research Asia from 2012 to 2013. Currently, her research interests include spatial-temporal data mining, bioinformatics, behavioral data mining and computational social science. She has received the best paper award in ICDM2013.



Xing Xie is currently a senior researcher in Microsoft Research Asia, and a guest Ph.D. advisor for the University of Science and Technology of China. He received his B.S. and Ph.D. degrees in Computer Science from the University of Science and Technology of China in 1996 and 2001, respectively. He joined Microsoft Research Asia in July 2001, working on spatial data mining, location-based services, social networks and ubiquitous computing. During the past years, he has published over 140 referred journal and conference papers, such as ACM Transactions on Intelligent Systems and Technology, ACM Transactions on the Web, ACM/Springer Multimedia Systems Journal, IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Mobile Computing, IEEE Transactions on Multimedia, etc. He has more than 50 patents filed or granted.