SmartTransfer: Modeling the Spatiotemporal Dynamics of Passenger Transfers for Crowdedness-Aware Route Recommendations

BOWEN DU and YIFENG CUI, Beihang University, China

YANJIE FU, University of Missouri-Rolla, USA

RUNXING ZHONG, Beihang University, China

HUI XIONG, Rutgers University, USA

In urban transportation systems, transfer stations refer to hubs connecting a variety of bus and subway lines and, thus, are the most important nodes in transportation networks. The pervasive availability of large-scale travel traces of passengers, collected from automated fare collection (AFC) systems, has provided unprecedented opportunities for understanding citywide transfer patterns, which can benefit smart transportation, such as smart route recommendation to avoid crowded lines, and dynamic bus scheduling to enhance transportation efficiency. To this end, in this article, we provide a systematic study of the measurement, patterns, and modeling of spatiotemporal dynamics of passenger transfers. Along this line, we develop a data-driven analytical system for modeling the transfer volumes of each transfer station. More specifically, we first identify and quantify the discriminative patterns of spatiotemporal dynamics of passenger transfers by utilizing heterogeneous sources of transfer related data for each station. Also, we develop a multi-task spatiotemporal learning model for predicting the transfer volumes of a specific station at a specific time period. Moreover, we further leverage the predictive model of passenger transfers to provide crowdedness-aware route recommendations. Finally, we conduct the extensive evaluations with a variety of real-world data. Experimental results demonstrate the effectiveness of our proposed modeling method and its applications for smart transportation.

CCS Concepts: • **Information systems** → **Spatial-temporal systems**; **Mobile information processing systems**; **Data mining**;

Additional Key Words and Phrases: Automated fare collection, transit behavior, spatiotemporal, crowdedness detection, route recommendation

ACM Reference format:

Bowen Du, Yifeng Cui, Yanjie Fu, Runxing Zhong, and Hui Xiong. 2018. SmartTransfer: Modeling the Spatiotemporal Dynamics of Passenger Transfers for Crowdedness-Aware Route Recommendations. *ACM Trans. Intell. Syst. Technol.* 9, 6, Article 70 (November 2018), 26 pages.
<https://doi.org/10.1145/3232229>

This research was supported in part by National Natural Science Foundation of China via Grant no. 51778033. This research was supported in part by Beijing Municipal Science and Technology Project no. Z171100000917016.

Authors' addresses: B. Du, Y. Cui, and R. Zhong, Advanced Innovation Center for Big Data and Brain Computing, Beihang University, No. 37 Xueyuan Road, Haidian District, Beijing 100191, China; emails: {dubowen, naskcyf, zhongrunxing}@buaa.edu.cn; Y. Fu (corresponding author), Missouri University of Science and Technology, Computer Science, 2430 D Lanes End, Rolla, MO 65409-0001; email: yanjiefoo@gmail.com; H. Xiong, Rutgers University, Management Science and Information System, Washington Park #1076 Newark, NJ 07102; email: hxiong@rutgers.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

2157-6904/2018/11-ART70 \$15.00

<https://doi.org/10.1145/3232229>

1 INTRODUCTION

Public transit systems (PTS), including buses and subways, have become the primary mode of transportation for daily commuters in many urban areas, such as New York City and Beijing. PTS actively seek opportunities to improve transportation efficiency and traveler experience because passengers consider not only travel time but also trip comfort (e.g., overcrowding). Understanding citywide passenger transfer patterns can be very helpful for the aforementioned two considerations of PTS. For example, modeling the spatiotemporal dynamics of passenger transfers can lay the foundation for developing smart route recommendations to avoid crowded lines.

Many existing systems, such as Google and Baidu maps, mostly suggest routes simply based on travel time or number of transfers, thereby typically ignoring trip crowdedness. Therefore, this highly necessitates a smart system that recommends routes based on both transportation efficiency and traveler experience. The pervasiveness of automated fare collection (AFC) has led to the accumulation of large-scale passenger transfer behavior. This transfer behavior, if properly analyzed, can be a source of rich intelligence to provide route recommendations with efficiency and comfort awareness.

However, two challenges arise regarding this goal: (i) how to forecast the volume of passenger transfers at a specific location and time period; and (ii) how to incorporate transfer volume forecasts into route recommendations with efficiency and comfort awareness.

First, we can extend autoregression [9, 25], interpolation [11, 17, 30], and Markov models to spatiotemporal dimensions to predict passenger transfer volumes; however, the performance of these methods can be improved through collective learning by incorporating fine-grained cross-domain data sources and capturing the spatiotemporal autocorrelations of passenger transfers within a transfer and across transfer stations. Therefore, we propose exploiting a spatiotemporal multi-task learning (MTL) strategy, and study the prediction of passenger transfers for each transfer station from two perspectives: (i) intra-station temporal dependencies; and (ii) inter-station spatial relatedness. Particularly, within a transfer station, even though the transfer volumes of the station change over time, the transfer volumes of the next time slot depend on the transfer volumes of the present time slot. This is called an intra-station temporal dependency. Across multiple transfer stations, if two stations are spatially close to each other, or two stations are far away from each other but share similar neighborhood profiles, it is quite likely that the two stations share similar patterns of transfer volumes. This is called inter-station spatial relatedness. Second, after forecasting the passenger transfer volumes of stations over time, we need to strategically incorporate the spatiotemporal volumes of passenger transfers into the smart route recommendation system to achieve a balance between time costs and trip crowdedness. The route recommender system consists of two stages. The first stage aims to generate candidate routes given the origin and destination locations. The second stage aims to measure and rank the utility of candidate routes in terms of both time costs and trip crowdedness.

Figure 1 shows the framework of our proposed system. Specifically, we first collect a variety of data, including bus/subway transaction records and geographic information, that are related to passenger transfers and develop reliable measurements for transfer volumes. Additionally, we identify and extract a variety of features that are related to passenger transfers, such as reachability, traffic capacity, direct and transfer ratio, passenger moving patterns affected by events and weather, and geo-social information of residential communities, such as the number of available facilities (e.g., shopping malls and residential communities) in the neighborhood, population density evolving at the same station, and deviation from the social norm (e.g., difference between individual behavior and the typical behavior of the population). Additionally, we develop an MTL framework to jointly model both intra-station temporal dependencies and inter-station spatial

Fig. 1. Proposed system framework.

relatedness to simultaneously perform temporal and spatial predictions of passenger transfers. The temporal prediction captures the dependencies between two temporally adjacent transfer volumes within a transfer station and forecasts transfer volumes in the future, whereas the spatial prediction captures the inter-station spatial relatedness across all stations and provides a more accurate prediction for transfer volumes at any location.

The prediction results of our model could be further used for smart route recommendation with efficiency and comfort awareness. The recommender system includes candidate route generation and route recommendation. Finally, we conducted intensive experiments with real-world data to evaluate our model and its applications for the aforementioned two problems. The experimental results clearly demonstrate the effectiveness of the proposed method and enhance our understanding of smart transportation.

2 DATA DESCRIPTION

Our study is based on a large-scale multi-source dataset collected in Beijing for the time period from August 2016 to October 2016. In this section, we provide an overview of the data.

In recent years, with advances in information technology and data processing capability, different types of data have been collected, such as transactional records [7], which are considered as an invaluable dataset for understanding passenger mobility patterns, in addition to urban dynamics. Specifically, each transaction record consists of a card identifier (ID), route number, state (boarding or exiting), name of a station, and timestamp. After data preprocessing, we prepared over 1.7 billion records for training, and found that the average number of daily travel records was approximately 8.5 million in Beijing, and transfer records that connected different stations represented at least 10% of all travel records, according to statistical analysis. The spatial distribution of passenger flow is shown in Figure 2, where the background map represents the area within the Fifth Ring Road in Beijing. The three-dimensional (3D) blue column diagram shows the passenger

Fig. 2. Passenger flow spatial distribution.

Table 1. Some Important Statistics for Our Data

Type	Scale description
AFC transactional records	Over 1.7 billion records of passenger’s mobility
Rental price	269 records of the rental price near the subway stations
Important events	2,690 records of some important events
Weather	92 weather records of 3 months
Road networks	1,685 bus lines and 6,406 bus stations 19 subway lines and 268 subway stations
Points of Interest	185,841 records of five types
Traffic zones	1,911 zones

flow volume of certain areas in Beijing, whereas the two-dimensional (2D) thermal diagram shows the passenger flow volume of general areas.

In addition to transactional records, six data sources were used in this study: rental price data, important events (2,690 records), weather data, road networks (1,685 bus lines, 6,406 bus stations, 19 subway lines, and 268 subway stations), points of interest (POI) (185,841 records), and traffic zones (1,911 zones). Some important statistics for our data are shown in Table 1.

3 FEATURE EXTRACTION

In this section, we identify and extract a variety of discriminative features from multi-source data. Our features consist of eight categories: transfer volume, transfer distance and time, accessibility,

Fig. 3. Distribution of transfer ratio and transfer time.

reachability, connectivity, functional zones, event and weather, and temporal-related features. Table 2 shows a complete description and statistics for these features, and Table 3 shows the mathematical notation used in the article.

To understand why passengers gather at a station, we define and identify the mobility patterns of each passenger. Specifically, the mobility pattern of a passenger is a triple that consists of the passenger's travel state, travel mode, and transfer mode. The travel state of a passenger can be arriving (ar) or leaving (le); the travel mode of a passenger can be traveling by bus or subway only during the trip, denoted by $m \in M = \{B \text{ or } R\}$; and the transfer mode of a passenger $tm \in TM = \{BR, RB\}$ can be transferring from bus to subway or from subway to bus during the trip. Formally, the definition of a passenger mobility pattern and the definition of transfer are given as follows:

Definition 1. R_i is the i th subway station, L_i is the neighboring area of the station, which considers R_i as the center, and r is the radius. Passengers who travel through R_i have the attributes of travel state, travel mode, and transfer mode.

Definition 2. Urban public transport transfer is a behavior activity for travelers to reach the destination, which includes the route changes between rail transit during the trip and the transformation of public transport during the trip [12]. In this article, *Transfer* refers to a passenger's changes from bus to subway or changes from subway to bus when he or she travels by public transportation.

We observed that the mobility patterns and trip purposes of passengers vary significantly over different time periods of a day. Therefore, we applied a temporal partition method, which was proposed by [19], on the passenger transfer records to summarize the statistics. The passenger transfer records were segmented into four major time slots, in terms of leaving times: 05:00–10:30, 10:30–16:30, 16:30–19:30, and 19:30–23:30. We segmented the time period into four because it covers four behaviors of crowd activity in a day that correspond to the time that people go to work, time people work, time people leave work, and time people experience nightlife.

Transfer Volume-Related Features. Most people need to travel by public transportation every day, especially in big cities such as Beijing. Many of them have to go through the traffic transfer scenario because of limited public transportation facilities in the city. Thus, it is significant to define some indices associated with passenger transfer flow at each station as the feature.

Figure 3(a) shows the transfer ratios at Tiantongyuanbei Station over different time periods. In Beijing, Tiantongyuanbei Station has a very high ratio of transfer volume to total traffic volume. The commute patterns of Tiantongyuanbei Station are as follows: During 05:00–10:30, 93% of

Table 2. Extracted Features of a Subway Station

Type	Feature	Description	Class	Mean	Median	Max	Min
Transfer volume	VX	Volumes of passenger arriving/leaving by bus/subway	Bin	324.81	228	7,713	1
			Rout	721.45	418	14,707	1
			Rin	726.02	422	18,774	1
			Bout	315.17	227	4,046	1
	TRX	Ratio of transferring who arriving/leaving by bus/subway	BRB	0.18	0.10	0.99	0
			BRR	0.09	0.06	0.99	0
			RBR	0.08	0.06	0.75	0
			RBB	0.18	0.11	0.99	0
Transfer distance and time	TDX	The walking distance that transfer required	BRtrDist	451.38	508	949	14
	RBtrDist		445.98	508	968	14	
	TTX	The time that transfer required	BRtrTime	326.69	318	1,485	1
			RBtrTime	533.34	562	1,523	1
Accessibility	CX	The number of connected zones	BRzoneFrom	12.60	10	135	0
			BRzoneTo	28.30	17	231	0
			RBzoneFrom	26.98	16	221	0
			RBzoneTo	11.57	9	89	0
Reachability	RX	How far can arriving/leaving to in unit time	BRreachFrom	17.97	18.06	82.47	3.60
			BRreachTo	16.68	18.13	47.56	3.60
			RBreachFrom	16.52	18.04	46.15	3.60
			RBreachTo	15.87	15.61	78.34	3.60
Connectivity	LX	How many bus/subway lines passing by	Blines	29.03	28.50	94	0
			Rlines	1.21	1	3	1
Functional zones	PrX	The price of rental nearby	Rents	81.22	80	182	29
	PX	The number of POI nearby	PoiSum	245.31	183.50	1781	0
	EX	The Entropy of POI class nearby	PoiEntropy	1.83	2.03	2.42	0
Event and weather	AX	Whether affected by event/weather	Event	0.09	1	1	0
			Weather	29.60	30	57	7
Temporal-related	TF	The time series features	SeriesFeature	0.44	0.44	1	0
	SF	The statics related features	Mean	0.07	0.04	1	0
			Std	0.06	0.05	1	0
			Min	0.04	0.02	1	0
			Max	0.08	0.04	1	0
			Median	0.07	0.03	1	0
			Prctile1	0.05	0.02	1	0
			Prctile2	0.07	0.04	1	0
			Skewness	0.42	0.37	1	0
			Kurtosis	0.27	0.22	1	0
			Autocorr	0.04	0.40	1	0
	FF	The frequency features including FFT and DWT	FFT	0.05	0.03	1	0
			DWT	0.06	0.03	1	0

Table 3. Mathematical Notation

Symbol	Description
o	The origin of a trip
d	The destination of a trip
e	The transfer station
t	The time period
m	An item of set M
tm	An item of set TM
j	An item of set J
ar	Travel state of arriving
le	Travel state of leaving
B	Travel mode of passenger travel by bus
R	Travel mode of passenger travel by subway
BR	Transfer mode of passenger transfer from bus to subway
RB	Transfer mode of passenger transfer from subway to bus
M	Travel mode set
TM	Transfer mode set
Y	The historical transfer passenger of a subway station in previous K day
R_i	The i th subway station
L_i	The neighboring area of R_i which treat R_i as the center and radius is r
J	The set of POI class
$dist(o, d)$	The travel distance from o to d
$time(o, d)$	The travel time from o to d
OD_t^m	The total set of passengers who travel from place o to d by travel mode m
OED_t^{tm}	Denote consecutive (o, d) where transfer at station e
$VX^m(t)$	The volume of passenger flow by travel mode m
$VX^{tm}(t)$	The volume of passenger flow by transfer mode tm
$TRX(o, d, e, t)$	The transfer ration
$TDX^{tm}(t, R_i)$	The transfer distance
$TTX^{tm}(t, R_i)$	The transfer time
$CX^{tm}(t, R_i)$	Denote how many different zones connect to the station R_i in time period t by transfer mode tm
$RX^{tm}(t, R_i)$	The travel distance in unit time of a passenger who leaves or arrives in the subway station R_i during time period t by tm transfer mode
$LX^m(R_i)$	The number of bus lines and subway lines that have bus stations and other subway stations installed near the subway station R_i
PX^{L_i}	The total number of POIs of the region L_i
$PX_j^{L_i}$	The number of POIs that belong to the category j
EX_{L_i}	The entropy of the POI class
$AX_t^{L_i}$	The number of events around the area L_i in time period t

passengers arrive at this station by bus to transfer to the Beijing subway. This ratio decreases as time goes on. Similarly, Figure 3(b) shows that the time that passengers spend on transfers gradually decreases as time goes on.

Formally, we let OD_t^m denote the total set of passengers who travel from place o to place d by travel mode m , and tuple (o, d) denotes travel without transfer, where $(o, d) \in OD_t^m$. Similarly, OED_t^{tm} denotes the total set of passengers who travel from place o to place d and transfer in tm

Fig. 4. Feature of accessibility and feature of transfer distance and time.

mode at station e , where triple $(o, e, d) \in OED_t^{tm}$, and it represents travel with a transfer, where t is the time period. More formally,

$$VX^m(t) = |\{(o, d) \in OD_t^m\}|, \quad (1)$$

$$VX^{tm}(t) = |\{(o, e, d) \in OED_t^{tm} : e = R_i\}|, \quad (2)$$

where $VX^m(t)$ and $VX^{tm}(t)$ denote the total volume of passenger flow from place o to place d in travel mode or transfer mode. Then, the transfer ratio, which denotes travel from o to d and transfer at station e in period t , can be expressed as

$$TRX(o, d, e, t) = \frac{VX^{tm}(t)}{VX^{tm}(t) + VX^m(t)}. \quad (3)$$

Transfer Distance and Time-Related Features. Intuitively, the average walking distance and time of transfers have a significant influence on passengers' choices. For instance, when given multiple transfer plans to a target destination, passengers usually choose a plan that consists of transfer stations with less time cost and short walking distances. Formally, we use $TDX^{tm}(t, R_i)$ and $TTX^{tm}(t, R_i)$ to denote transfer distance and time, respectively. These features are calculated using the median rather than mean value because median values are more robust to outliers. Figure 4(b) shows the median transfer distance and median transfer time of Tiantongyuanbei Station on August 10, 2016 where $BRtrTime$ and $RBtrTime$ denote the transfer time feature, BR and RB denote the transfer modes, and $BRtrDist$ and $RBtrDist$ denote the transfer distance feature. The unit of transfer time is the second and that of transfer distance is the meter.

Accessibility-Related Features. Accessibility is another important factor when choosing transfer stations. Accessibility represents whether a subway station has strong connectivity in transportation networks, particularly how many traffic zones a subway station can reach. For instance, there are 1,911 traffic zones in Beijing, and starting from Tiantongyuan Station, passengers can reach 551 of these zones. According to our collected data, in Beijing, passengers traveling from a subway station can reach 284 traffic zones on average. We use $CX^{tm}(t, R_i)$ to denote the number of different zones connected to station R_i in time period t , where connect means the number of different zones from which passengers can come to station R_i in time period t by transfer mode tm or the number of different zones from which passengers arrive from station R_i in time period t by transfer mode tm . Figure 4(a) shows the accessibility feature of 10 selected subway stations in Beijing, and $BRzoneFrom$, $BRzoneTo$, $RBzoneFrom$, and $RBzoneTo$ represent the four specific categories of $CX^{tm}(t, R_i)$. Consider $BRzoneFrom$ and $BRzoneTo$, for example, where $RBzoneFrom$ and

$RBzoneTo$ are the same. $BRzoneFrom$ denotes the number of different zones from which passengers can come to station R_i by transfer mode BR , and $BRzoneTo$ denotes the number of different zones from which passengers arrive from station R_i by transfer mode BR . YBR denotes the volume of passenger flow that transfers by BR mode at 10 stations and YRB denotes the volume of passenger flow that transfers by RB mode at 10 stations. Additionally, in Figure 4(a), the y -axis on the left-hand side corresponds to the bar graph and the y -axis on the right-hand side corresponds to the line graph.

Reachability-Related Features. We devise a feature $RX^{tm}(t, R_i)$ to measure the travel distance in unit time of a transfer passenger who leaves or arrives at subway station R_i during time period t by tm transfer mode. Formally,

$$RX^{tm}(t, R_i) = \frac{\sum \frac{dist(o,d)}{time(o,d)}}{|VX^{tm}(t)|}. \quad (4)$$

Connectivity-Related Features. The more bus/subway lines that a station can connect, the easier passengers can travel. This feature reflects the convenience and attractiveness of a station. Formally, we use $LX^m(R_i)$ to denote the number of bus lines and subway lines that have bus stations and other subway stations near subway station R_i .

Functional Zone-Related Features. Functional zones reflect the travel purposes of passengers and support their different needs regarding socioeconomic activities. By measuring the number of facilities (marked as a POI in map data), that is, the number of each POI category that passengers can access, rental price in area L_i , and diversity (represented by entropy) of the POI category, we can estimate if a functional zone can support the demands of passengers, such as living demands (e.g., restaurants and supermarkets) and education demands (e.g., middle schools and universities). Formally, we use PX^{L_i} to denote the total number of POIs of region L_i , and $PX_j^{L_i}$ to denote the number of POIs that belong to category j . Specifically, we classify POIs into six categories as $j \in J = \{Live, Business, Entertainment, Education, Healthcare, Life Services\}$ and the entropy of the POI categories is defined as

$$EX_{L_i} = - \sum \frac{PX_j^{L_i}}{PX^{L_i}} \times \log \left(\frac{PX_j^{L_i}}{PX^{L_i}} \right). \quad (5)$$

Event and Weather-Related Features. Events and weather have a significant effect on passenger flow; they affect the decision-making process of choosing a travel mode, that is, taxi or subway.

For example, the yellow curve in Figure 5(a) shows a concert that was held near Xitucheng Station and thus attracted a large number of passengers in that area at that time. The blue curve in Figure 5(b) shows that passenger flow decreased during a holiday, that is, the Mid-Autumn Festival, which is significantly different from regular days. Formally, we measured the number of events around area L_i in time period t to estimate the deviation degree of passenger flow, which is denoted by $AX_t^{L_i}$.

Temporal-Related Features. To capture the features from a temporal viewpoint, we consider historical passenger transfer flow Y at each subway station in the past K days. We denote time series $[Y_{[t-k]}, Y_{[t-k+1]}, \dots, Y_t]$ by TF . To comprehensively understand the temporal features, we extract features based on the following method. First, we extract the statistical features from the time series, denoted by SF : *mean, variance, median, upper quarter quantile, lower quarter quantile, maximum, minimum, skewness, kurtosis, and auto correlation*. Second, to represent the temporal features accurately, we extract features from the frequency domain, denoted by FF : Fast Fourier transform (FFT) and Discrete Wavelet transform (DWT).

Fig. 5. Example of passenger flow affected by events in Xitucheng.

4 MULTI-TASK TRANSFER PASSENGER FLOW PREDICTION (MTPFP)

In this section, we introduce how to predict the passenger transfer volumes at each subway station in different time slots using a spatialtemporal multi-task learning method.

Let $Y \in \mathbb{R}^{N \times T}$ denote the passenger transfer flow in N subway stations during T days. $X = [VX, TRX, TDX, TTX, CX, RX, LX, PrX, PX, EX, AX]$, where $X \in \mathbb{R}^{N \times T \times M}$, denotes the extracted features for N subway stations and T time slots. M denotes the number of spatial features. $Q = [TF, SF, FF]$, where $Q \in \mathbb{R}^{N \times T \times K}$, denotes the temporal features, which capture the characteristics related to the previous K days of transfer passenger flow Y .

4.1 Objective Function

Our goal is to use the previously proposed features to predict passenger flow of multiple stations at the same time, so our objective function needs to take all types of features into consideration. For this purpose, we get the *formula(6)*. The first part of *formula(6)* introduces spatial features and it obtains the predicted value of passenger flow through the learning of spatial features. The second part of *formula(6)* introduces the time series weight term and the spatial feature term, and the combination of the two terms can achieve the purpose of constraining the training parameters so that different categories of features can play a role in training at the same time. By adding the two parts, we get the basic optimization model which has the ability of learning spatial characteristic and time characteristic. So the optimization objective is to minimize the sum of squared error between the predicted passenger flow based on spatial features and the benchmark passenger flow, and the sum of square error between the predicted passenger flow based on spatial features and the predicted passenger flow based on temporal features:

$$J(W, U) = \frac{\alpha_1}{2} \sum_{r=1}^N \|Y_r^T - X_r W_r^T\|_F^2 + \frac{\alpha_2}{2} \sum_{r=1}^N \|Q_r U_r^T - X_r W_r^T\|_F^2, \quad (6)$$

where $\|\cdot\|_F$ is the Frobenius norm, T denotes a matrix transpose, $W \in \mathbb{R}^{N \times M}$ and $U \in \mathbb{R}^{N \times K}$ are the parameters that we need to learn in our model, and α_1, α_2 are the hyperparameters.

In practice, the stations with similar neighborhood profiles should demonstrate very high spatial similarities, such as Zhichunlu subway station and Wudaokou subway station, which have

similar neighborhoods. For example, there are many schools around the Zhichunlu subway station, including primary schools, middle schools, and universities, as does the Wudaokou subway station; there are about 10 bus stations within 1,000 meters of Zhichunlu subway station, and also for Wudaokou subway station; there are many restaurants and entertainment venues around both Wudaokou subway station and Zhichunlu subway station. So the two stations are similar in space and our features also confirm this point. To incorporate such spatial autocorrelation into our objective function, we add the *Laplacian matrix* regularizer into our model to make it more reasonable.

Specifically, we define the regularizer $S_{ij} = \frac{X_i \cdot X_j}{||X_i||_F^2 \times ||X_j||_F^2}$ as the spatial similarity between stations i and j , where $S \in \mathbb{R}^{N \times N}$ is a symmetric matrix. W is the weight matrix of X .

Moreover, to obtain good performance in the prediction stage, we incorporate *Lasso* and *Ridge* regularization terms into our model. *Lasso* regularization $||W||_{2,1}$ can help to select important features by regularizing their parameters to zero. We use *Ridge* regularization $||W||_F^2$ and $||U||_F^2$ to reduce the variances of the parameters in the prediction. And the detailed process of *Lasso* handled in the gradient computation is as follows. The role of *Lasso* is to limit the parameter values of the model, so as to reduce the complexity of the model and effectively prevent the occurrence of overfitting in the objective function. In the process of gradient calculation, the gradient change caused by the *Lasso* regularization term takes 1 when $W_j > 0$, and takes -1 when $W_j < 0$ (W_j is a learning parameter in W), so when $W_j > 0$, it will subtract a positive number to reduce itself, and when $W_j < 0$, it will subtract a negative number to increase itself, therefore, both of these operations will make the learning parameter W_j close to 0, which is the reason why *Lasso* regularization can make the weight of the model so sparse; finally, the model will be simple. Then, our final optimization objective function is

$$\begin{aligned} J(W, U) = & \frac{\alpha_1}{2} \sum_{r=1}^N ||Y_r^T - X_r W_r^T||_F^2 \\ & + \frac{\alpha_2}{2} \sum_{r=1}^N ||Q_r U_r^T - X_r W_r^T||_F^2 + \lambda_1 ||W||_{2,1} \\ & + \frac{\lambda_2}{2} ||W||_F^2 + \frac{\lambda_3}{2} ||U||_F^2 + \frac{\rho}{2} \sum_{i=1}^N \sum_{j=1}^N S_{ij} ||W_i - W_j||_F^2, \end{aligned} \quad (7)$$

where $\lambda_1, \lambda_2, \lambda_3, \rho$ are the hyperparameters.

4.2 Solving the Optimization Problem

To help the calculation, we mathematically transform the *Laplacian* term into another form: $\sum_{i=1}^N \sum_{j=1}^N S_{ij} ||W_i - W_j||_F^2 = 2tr(W^T L W)$. $L = D - S$, where D is a diagonal matrix for which $D_{ii} = \sum_j S_{ij}$. Finally, our objective function is defined as follows:

$$\begin{aligned} \arg \min_{W, U} J(W, U) = & \frac{\alpha_1}{2} \sum_{r=1}^N (||Y_r^T - X_r W_r^T||_F^2) \\ & + \frac{\alpha_2}{2} \sum_{r=1}^N (||Q_r U_r^T - X_r W_r^T||_F^2) + \lambda_1 ||W||_{2,1} \\ & + \frac{\lambda_2}{2} ||W||_F^2 + \frac{\lambda_3}{2} ||U||_F^2 + \rho tr(W^T L W). \end{aligned} \quad (8)$$

Table 4. Mathematical Formula Descriptions

Formula	Description
<i>formula(6)</i>	The basic objective function
<i>formula(7)</i>	The basic objective function with three basic regularization terms and a Laplacian matrix regular term
<i>formula(8)</i>	The final objective function
<i>formula(9)</i>	Calculate the learning parameter W_r based on coordinate descent method
<i>formula(10)</i>	Calculate the learning parameter U_r based on coordinate descent method
<i>formula(11)</i>	Calculate the partial derivatives of the objective function to the parameter W_r
<i>formula(12)</i>	Calculate the partial derivatives of the objective function to the parameter U_r

To optimize this objective function, we search the parameters W and U based on the coordinate descent method:

$$W_r = W_r - \sigma_w \frac{\delta J(W, U)}{\delta W_r}, \quad (9)$$

$$U_r = U_r - \sigma_u \frac{\delta J(W, U)}{\delta U_r}, \quad (10)$$

$$\begin{aligned} \frac{\delta J(W, U)}{\delta W_r} = & -\alpha_1 (Y_r^T - X_r W_r^T)^T X_r, \\ & -\alpha_2 (Q_r U_r^T - X_r W_r^T)^T X_r + \lambda_2 W_r + \rho((L + L^T)W)_r, \end{aligned} \quad (11)$$

$$\frac{\delta J(W, U)}{\delta U_r} = \alpha_2 (Q_r U_r^T - X_r W_r^T)^T Y_r^T + \lambda_3 U_r. \quad (12)$$

Table 4 lists the descriptions of each equation.

ALGORITHM 1: Objective Function Optimization

Input: X_r is the feature of station r ; Y_r is the target value of station r ; Q_r is the temporal feature of station r ; L is the similarity constraint matrix; $\alpha_1, \alpha_2, \lambda_1, \lambda_2, \lambda_3, \rho$ are the hyperparameters of the model; *epoch* is the maximum number of iterations

Output: W, U

```

1: Initialize  $W, U, \sigma_w, \sigma_u, \text{Error}$ 
2:  $JJ[0] \leftarrow J(W, U, X_r, Y_r, Q_r, L, \alpha_1, \alpha_2, \lambda_1, \lambda_2, \lambda_3, \rho)$ 
3: for  $i$  from 1 to epoch do
4:   // The partial derivative of  $J$  to  $W$ 
5:    $dW \leftarrow \text{dJdW}(W, U, X_r, Y_r, Q_r, L, \alpha_1, \alpha_2, \lambda_1, \lambda_2, \lambda_3, \rho)$ 
6:    $W \leftarrow W - \sigma_w \times dW$ 
7:   // The partial derivative of  $J$  to  $U$ .
8:    $dU \leftarrow \text{dJdU}(W, U, X_r, Y_r, Q_r, L, \alpha_1, \alpha_2, \lambda_1, \lambda_2, \lambda_3, \rho)$ 
9:    $U \leftarrow U - \sigma_u \times dU$ 
10:   $JJ[i] \leftarrow J(W, U, X_r, Y_r, Q_r, L, \alpha_1, \alpha_2, \lambda_1, \lambda_2, \lambda_3, \rho)$ 
11:  if  $\text{abs}(JJ[i] - JJ[i-1]) < \text{Error}$  then
12:    break
13:  end if
14: end for
15: return  $W, U$ 

```

5 ROUTE RECOMMENDATION

In this section, we develop a method to provide crowdedness-aware route recommendations (CARR) based on the transfer passenger flow prediction results. Most people consider time cost first when they travel. However, with the increase of transportation modes and route choices, people try to achieve a tradeoff between time and comfort. Accordingly, we provide a CARR model, which can recommend routes for passengers, and considers both travel time and crowdedness according to different requirements.

5.1 Route Generation

Because of the richness and complexity of PTS, passengers can search and identify many candidate routes using online map services at the present time. However, these routes have only several minutes' time difference but significantly vary in terms of crowdedness. Formally, the networks of PTS are defined as

$$\begin{cases} PTS = \{L, S, A, T\}, \\ l_i = \{s_j^i | j = 1, 2, \dots, n\}, \\ A = \{< s_j, s_k > | s_j, s_k \in S_i\}, \\ T = \{< s_j^i, s_{\hat{j}}^{\hat{i}} > | s_j^i \in S_i, s_{\hat{j}}^{\hat{i}} \in S_{\hat{i}}\}, \end{cases} \quad (13)$$

where L denotes the set of all routes and S denotes the set of all stations. $S_i \in S$ denotes the station set of line $l_i \in L$ and s_j^i denotes the j th station of line l_i . A denotes the accessible station pair set and is used to record the station pairs that have a direct route between them. T denotes the transfer station pair set, which indicates that passengers can transfer their travel between lines l_i and $l_{\hat{i}}$. The time cost is determined from statistics by analyzing historical data [33].

Then, we obtain the reachable routes by searching the network based on the breadth-first search algorithm. The algorithm description is given as follows:

Route Generation Description

- (1) Construct two station sets, $S1$ and $S2$, which represent the accessible station pair set of the origin station and destination station, respectively.
- (2) Iteratively select station s_k from $S1$ and construct a route set such that the origin station can directly access station s_k , denoted by candidate route set CRT . RT stores the routes from which the passenger can travel from the origin station to the destination station.
- (3) Assess each candidate route crt in CRT : if $S2$ contains the last station of crt , then add crt into RT . Otherwise, add all routes that can be reached from crt by considering one more line into $nextCRT$. Then add $nextCRT$ into CRT after removing all loop routes inside it.
- (4) Repeat the above process until the search is complete.
- (5) Return route set RT .

5.2 Crowdedness Awareness Route Recommendation

We introduce a method to recommend routes with an awareness of crowdedness and travel time. For all candidate routes, we first define a preference function $Pre(rt)$ as follows:

$$Pre(rt) = \frac{\beta^2 + 1}{\beta^2 \frac{1}{T(rt)} + \frac{1}{C(rt)}} + \Omega(rt), \quad (14)$$

where $Pre(rt)$ is the weighted harmonic mean of $T(rt)$ and $C(rt)$, and β is a weight parameter. When the value of β increases, the result of $Pre(rt)$ is determined mainly by the value of $T(rt)$; otherwise, it is determined mainly by the value of $C(rt)$, where $rt \in RT$ is one route in route set

ALGORITHM 2: Route Generation

Input: L, S, A, T are the parameters of PTS; o is the origin station; d is the destination station

Output: *Route*

```

1: Initialize candidate route set  $CRT$ 
2: Initialize valid route set  $RT$ 
3:  $S1 \leftarrow \text{getDirectAccessStation}(L, S, A, o)$ 
4:  $S2 \leftarrow \text{getDirectAccessStation}(L, S, A, d)$ 
5: for  $s$  in  $S1$  do
6:    $crt \leftarrow \text{getRoute}(o, s)$  // route from  $o$  to  $s$ 
7:   Add  $crt$  into  $CRT$ 
8:   while  $CRT$  is not empty do
9:      $crt \leftarrow CRT.pop()$ 
10:     $ls \leftarrow crt.getLastStation()$ 
11:    if  $S2$  contains  $ls$  then
12:      Add  $crt$  into  $RT$ 
13:    else
14:      // from  $o$  to  $ls$  and then transfer to another line
15:       $nextCRT \leftarrow \text{getOneTransfer}(L, S, A, T, crt, o, ls)$ 
16:      Remove the loop routes in  $nextCRT$ 
17:      Add  $nextCRT$  into  $CRT$ 
18:    end if
19:  end while
20: end for
21: return Route

```

RT. $T(rt) \in [0, 1]$ is the degree of saved time, which is calculated as follows:

$$T(rt) = \begin{cases} 1 - \frac{t_{rt} - t_{min}}{\Delta t}, & t_{rt} - t_{min} \leq \Delta t; \\ 0, & t_{rt} - t_{min} > \Delta t, \end{cases} \quad (15)$$

where t_{rt} is the travel time of route rt , t_{min} is the travel time of the fastest route, and Δt is the time threshold that determines whether a route can be considered. When $T(rt) = 1$, the rt time cost minimizes time t_{min} , and when $T(rt) = 0$, the rt time cost is at least Δt more than t_{min} . Then, we combine the transfer passenger flow prediction result as follows:

$$Z(rt) = \frac{1}{N_{PS}} \sum \frac{Y_{ps}}{norm_{ps}} + \mu \sum P(tr) \frac{Y_{tr}}{norm_{tr}}, \quad (16)$$

$$C(rt) = \frac{\sigma}{Z(rt)}, \quad P(rt) = \frac{trTime(tr)}{\sum trTime(tr)}, \quad (17)$$

where $C(rt)$ is the degree of comfort. ps is the station that route rt passes through and tr is the station from which route rt has transferred. $norm_{ps}$ and $norm_{tr}$ are the daily average passenger flow of stations ps and tr , respectively. N_{PS} is the number of stations that route rt passes through, $trTime(tr)$ is the transfer time cost at station tr , and Δt , σ , and μ are superparameters to balance the model.

$\Omega(rt)$ is the penalty terms of the route rt time cost at transfer.

Finally, we recommend the routes sorted by preference $Pre(rt)$.

Table 5. Parameter Settings of MTPFP

Setting	α_1	α_2	β_1	β_2	β_3	ρ
MTPFP@TLRS	1.0	0.5	0.001	0.015	0.01	0.02
MTPFP@TLR	1.0	0.5	0.001	0.015	0.01	0.00
MTPFP@T	1.0	0.5	0.000	0.000	0.00	0.00
MTPFP@N	1.0	0.0	0.000	0.000	0.00	0.00

6 EXPERIMENTAL RESULTS

6.1 Experiment Setup

Evaluation Metrics. To demonstrate the effectiveness of our method, we compared it with other algorithms in terms of the root-mean-square error (RMSE) and mean absolute error (MAE), which are defined as $RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}$ and $MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$, respectively, where y_i and $\hat{y}_i$ denote the benchmark volume and predicted volume of passenger transfers for station i , respectively, and n is the number of stations. We also use the decision tree method to calculate the Gini impurity criterion of different features, which is defined as $Gini(D, A) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$, where $Gini(D) = 1 - \sum_{k=1}^K (\frac{|C_k|}{|D|})^2$ and D is the sample set, D_1 and D_2 are two subsets of D after division by feature A , and C_k is the k th class subset of D , where K is the number of classes. Moreover, to reflect the importance of different features in forecasting tasks, we use the characteristic important factor (CIF) to evaluate the importance of each feature, where the CIF is defined as $CIF = 1 - Gini(D, A)$.

Implementation Details. Table 1 describes the data types that we need for our experiments. The AFC transactional records which including bus travel records and subway travel records are mainly collected by the automated fare collection (AFC) systems, and other types of data such as weather data are collected by web crawler or some public official information. The specific amounts of data used in our experiment are 45 million effective bus and subway travel records, 269 rent price information near subway stations, 2,500 useful event information, 92 weather records, 1,685 bus lines and 6,406 bus stations, 19 subway lines and 268 subway stations, 185,000 points of interest record, and 1,911 traffic zones. The time range of these data is from August 1 to October 31, which constitute our final experimental dataset. We split the dataset into the training set, validation set, and test set for three time periods: (i) August 01, 2016 to October 03, 2016; (ii) October 04, 2016 to October 17, 2016; and (iii) October 18, 2016 to October 31, 2016, respectively. All the experiments were conducted on a 3.6GHz $\times$ 8 Core CPU and 32.0GB main memory PC using Python 2.7, Java SE 1.8, and Matlab R2014a on a Windows 7 64-bit system. We called our method multi-task transfer passenger flow prediction (MTPFP).

6.2 Evaluation of Passenger Transfer Prediction

Study of the Regularization Terms. To demonstrate the effectiveness of the regularization terms used in our method, we designed several variants containing different combinations of regularization terms. As shown in Table 5, MTPFP@TLRS is the setting of our proposed model that combines temporal regularization terms, lasso regularization terms, ridge regularization terms, and spatial regularization terms; MTPFP@TLR only considers temporal regularization terms, lasso regularization terms, and ridge regularization terms; MTPFP@T only considers temporal regularization terms; and MTPFP@N only considers the least square error term. Figure 6(a) shows the performance comparison of the four MTPFP variants. First, MTPFP@TLRS is the best among all four variants. By comparing MTPFP@TLRS with MTPFP@TLR, we observe that when the number of stations increases, MTPFP@TLRS outperforms MTPFP@TLR significantly. This

Fig. 6. Performance evaluation of MTPFP. We calculated the RMSE and MAE of predicting the top 10, 20, and 268 stations, which are sorted by transfer flow.

observation validates the effectiveness of spatial similarity regularization. Second, comparing MTPFP@TLR with MTPFP@T, MTPFP@TLR outperforms MTPFP@T. This observation validates the effectiveness of lasso and ridge regularization. Third, comparing MTPFP@T with MTPFP@N, MTPFP@T demonstrates better performance than MTPFP@N. This observation demonstrates the effectiveness of temporal features. Additionally, from Figure 6(d), we can determine similar results. To summarize, Figure 6(a) and Figure 6(d) validate the effectiveness of MTPFP@TLRS.

Overall Performance. To demonstrate the effectiveness of our method, we compared it with the following baseline algorithms: (1) **LR**: trains a linear regression model to predict passenger transfer flow. (2) **LASSO**: trains a lasso regression model (i.e., LR with the L1 regularization). (3) **RIDGE**: exploits a ridge regression model (i.e., LR with the L2 regularization). (4) **KNN**: trains two KNN regression models by exploiting two different distances: uniform and Euclidean denoted by **KNN_unif** and **KNN_dist**, respectively. (5) **SVM**: trains three support vector machine-based regression methods by exploiting linear, poly, and RBF kernels denoted by **SVM_linear**, **SVM_poly**, and **SVM_rbf**, respectively. (6) **DT**: exploits a decision tree regression method. (7) **RF**: exploits a random forest regression method. (8) **ARMA**: exploits an autoregression and moving average regression method that is a famous time series prediction model. (9) **LR**: trains a basic Logistic regression model to predict passenger transfer flow. (10) **OLS**: trains an Ordinary Least Squares model to predict passenger transfer flow. (11) **MVR**: exploits a multi-variant regression method. Figure 6(b) and (e) show the performance comparison of our method and the 14 baseline algorithms. As can be observed, our method achieves the best performance compared with the baseline algorithms. Moreover, when the number of stations to be predicted increases, the accuracy of our method increases. The accuracy comparison demonstrates the effectiveness of combining MTL, spatial regularization, and temporal regularization.

Study of Joint and Separate Prediction for Weekdays and Weekends. We report the performance comparison of predicting passenger transfers for two strategies: (i) MTPFP@WWS: We trained MTPFP for a weekday training set or weekend training set to predict weekday or weekend

Fig. 7. Feature evaluation of MTPFP. We used three strategies to calculate the CIF of predicting the top 10, 20, and 268 stations, which are sorted by transfer flow.

passenger transfers, respectively. This was precisely our proposed strategy in the previous experiments. (ii) MTPFP@WWT: We trained MTPFP by combining both weekday and weekend training sets to predict weekdays and weekends together. Figure 6(c) and (f) show that MTPFP@WWS (separate prediction) performs better than MTPFP@WWT (joint prediction). This can be explained by the fact that weekdays and weekends have different passenger transfer patterns because people commute to work and get home from work on weekdays, whereas people commute for entertainment on weekends. To summarize, the strategy of predicting weekday and weekend passenger transfers individually is better than the joint prediction of both weekdays and weekends.

Study of the Importance Degree of Different Features. We report the importance of different features by predicting passenger transfers for three strategies: (i) MTPFP@SO: We only used the spatial features to train MTPFP. (ii) MTPFP@TO: We only used the temporal features to train MTPFP. (iii) MTPFP@ST: We used the spatial features and temporal features together to train MTPFP. Figure 7(a)–(c) show three types of results regarding MTPFP@SO. From the curves, we can determine that with an increase in the number of stations, TRX becomes a key feature in determining the value of the forecast, which can be explained by the fact that the more stations there are, the more likely a passenger will be able to transfer. We also find that VX, RX, and CX have a major effect on predictions. Figure 7(d)–(f) show the second strategy. It is clear that the feature SF plays a key role in the prediction results, which confirms the correctness of our temporal feature extraction. Additionally, when combining the two types of features for prediction, we obtain the

Fig. 8. Effectiveness evaluation of CARR.

Table 6. Description of Candidate Routes

Route ID	Model	Stations
rt01	R9 > R4 > R2(N)	S > A > B > E
rt02	R9 > R6 > R2(N)	S > C > D > E
rt03	R9 > R1 > R2(N)	S > M > F > E
rt04	R9 > R1 > R2(W)	S > M > G > E
rt05	R7 > R4 > R2(W)	S > H > I > E
rt06	R7 > R5 > R2(W)	S > J > K > E
rt07	R7 > R5 > R2(N)	S > J > L > E

results in Figure 7(g)–(i). From the curves that used MTPFP@ST, we find that the temporal features are dominant in predictions compared with the spatial features, and some spatial features also have a particular effect on the experimental results. These results prove the importance of historical data and the necessity of some spatial features in a prediction. To summarize, both temporal and spatial features have an effect on the accuracy of the prediction result, and choosing two types of features simultaneously is a good choice for our model.

6.3 Route Recommendations

The public transportation system of Beijing is one of the largest systems in terms of network density, crowdedness, and travel time. It has 1,685 bus lines, 6,406 bus stations, 19 subway lines, and 268 subway stations. In our route recommender system, given an origin and destination point, we can generate k candidate routes. For instance, as shown in Figure 8(a), there are seven candidate routes from Beijingxizhan to Dongzhimen. The details of the seven candidate routes are provided in Table 6.

We defined a preference by fusing both time costs and trip crowdedness. Figure 8(b) shows the preference scores of each candidate route. Based on the scores, we recommend route rt06 because it has the best balance between $T(rt)$ and $C(rt)$, as shown in Figure 8(c). According to the recommendation of rt06, passengers could take subway line R7 from Station S to Station J, transfer to subway line R5 for Station K, and finally, transfer to subway line R2 for Station E. Next, we compared our method called CARR with several baseline methods.

Comparison with Time Cost-Based Recommendation. An alternative approach to recommending routes is to simply rank candidate routes by time cost. We call this method TTRR. Table 7 shows that CARR recommends route rt06, whereas TTRR recommends route rt02. A close check of rt06 and rt02 determines that there is only 5 minutes difference between rt06 and rt02 in terms of the historical travel time: rt06 costs 53 minutes, on average, whereas rt02 cost 48 minutes, on

Fig. 9. Real passenger flow images in a subway station during non-peak hours and peak hours.

Table 7. Routes Recommended by CARR and TTRR

Rank	CARR	TTRR
1	rt06	rt02
2	rt05	rt01
3	rt04	rt04
4	rt01	rt05
5	rt02	rt06
6	rt07	rt03
7	rt03	rt07

average. However, Figure 8(c) shows that rt06 is much more comfortable in terms of crowdedness because CARR achieves a balance between time and comfort.

Study of Time-Aware Crowdedness. Because our method recommends routes in terms of spatiotemporal dynamics of passenger transfers, it can capture crowdedness varying over time, and dynamically recommend routes for different time periods. Specifically, we compared two time periods: (i) 05:10–10:30 (morning rush hour) and (ii) 10:30–16:00 (working hours with fewer passenger transfer).

According to the definition of $Pre(rt)$, when $\beta > 1.0$, time costs play a more important role, whereas, when $\beta < 1.0$, comfort plays a more important role. As shown in Table 8, when $\beta = 1.2$, the recommended routes in 05:10–10:30 and 10:30–16:00 are different. The proposed method recommends rt05 in 05:00–10:30 and rt02 in 10:30–16:00 because in morning rush hour, that is, 05:00–10:30, rt05 is more comfortable and only costs three more minutes than rt02, on average. Whereas during daily working hours, that is, 10:30–16:00, our method recommends the fastest

Table 8. Ranking of Different Parameters

Rank	05:00–10:30		10:30–16:30	
	$\beta = 0.8$	$\beta = 1.2$	$\beta = 0.8$	$\beta = 1.2$
1	rt06	rt05	rt06	rt02
2	rt05	rt06	rt02	rt06
3	rt04	rt01	rt05	rt05
4	rt01	rt04	rt07	rt04
5	rt02	rt02	rt04	rt01
6	rt07	rt07	rt01	rt07
7	rt03	rt03	rt03	rt03

Table 9. Results Compared with Other Methods

Our Model	Google	Baidu	Sogou	AMap	Tencent
rt06	rt05	rt01	rt02	rt05	rt01
rt05	rt06	rt05	rt01	rt01	rt05
rt04	rt04		rt04	rt02	
rt01	rt02				
rt02					
rt07					
rt03					

route rt02 because passenger flows are much lower during daily working hours. When $\beta = 0.8$ (comfort plays a more important role), the recommended route for 05:00–10:30 is different and the proposed method recommends rt06, which is different from the recommended result (route rt05) when $\beta = 1.2$. As can be observed from Figure 8(c), rt06 is more comfortable, whereas rt05 is faster. To summarize, our method can recommend routes dynamically and flexibly with an awareness of time-evolving crowdedness.

Comparison with Industry Route Recommendation Systems. In Table 9, we compared our method with some widely used map service providers: Google Map (<https://www.google.com/maps>), Baidu Map (<http://map.baidu.com>), Sogou Map (<http://map.sogou.com>), AMap (<http://ditu.amap.com>), and Tencent Map (<http://map.qq.com>). In the morning rush hour (05:00–10:30), Google, Baidu, AMap, and Tencent mostly recommend either rt01 or rt05; Sogou Map recommends rt02; and our method recommends rt06. A close check determines that rt06 achieves the best balance between time costs and crowdedness, as shown in Figure 8(c). To illustrate the results visually, we provided an NDCG scatter plot for each method. In the NDCG scatter plot, the x -axis represents the NDCG of a recommendation based on the benchmark route rankings sorted by $T(rt)$, and the y -axis represents the NDCG of a recommendation based on the benchmark route rankings sorted by $C(rt)$. A good recommendation method is expected to appear in the top right corner of the NDCG scatter plot. Figure 8(d) shows that our method achieves the best balance between travel time and crowdedness.

Validation of the Route Recommendation Results. We conducted a verification experiment on the routes recommended by CARR and TTRR. First, we divided the experimenters into seven groups and then set two time periods, 15:00–16:00 and 17:35–18:35, as off-peak and peak periods, respectively. Next, different groups of researchers took the subway from Dongzhimen Station to Beijing West Station according to a given route based on the preset time period. Simultaneously,

Table 10. Experimental Results of the Recommended Routes

Linecode	15:00–16:00		17:35–18:35	
	ComfortLevel	TimeCost(min)	ComfortLevel	TimeCost(min)
rt01	♣♣♣♣♣	35	♣♣♣	38
rt02	♣♣♣♣♣	40	♣♣♣	37
rt03	♣♣	50	♣	51
rt04	♣♣♣	47	♣♣	47
rt05	♣♣♣♣♣	42	♣♣♣♣	42
rt06	♣♣♣♣♣	43	♣♣♣♣	43
rt07	♣♣	51	♣	52

Fig. 10. Screenshot of the prototype system’s route recommendation function.

crowdedness and travel time were recorded during the trip. We conducted the experiment twice. Detailed experimental results are shown in Table 10 and Figure 9. We defined a hierarchical index in Table 10 to assess the comfort degree of different routes, and this index was acquired using different groups of experimenters’ true feelings. It is divided into five levels, such that the more “♣”, the higher the comfort degree; similar comfort feelings has the same level. Additionally, Figure 9 shows the actual passenger flow in the subway station for different routes and periods. Figure 9(a)–(c) represent the real passenger flow during non-peak hours, from which we can find that Figure 9(a) demonstrates the best scenario for passenger flow in the subway station, Figure 9(b) follows, and Figure 9(c) is the worst, which is consistent with Table 10 and our model, and our results are even more consistent during peak hours. All these real results validate the rationality of CARR.

7 PROTOTYPE SYSTEM

The prototype system, CARR, was implemented by Java language, Bootstrap frame, and SparkQL technique. Figure 10, Figure 11, and Figure 12 are screenshots of the prototype system. The system has the following basic components:

Fig. 11. Screenshot of the prototype system's line comfort degree display function.

Fig. 12. Screenshot of the prototype system's passenger flow status display function.

Origin and Destination Input. The input box is for the user to input the origin and destination. Additionally, it provides a button for the user to switch between the origin and destination. As shown in Figure 10, we entered the origin as Beijing West Railway Station and destination as Dongzhimen.

Option. As mentioned in the previous section, our system is crowdedness aware, and sensitive to different parameter settings and different time slots. Additionally, the user can sort the routes

by preference or travel time. As shown in Figure 10, we set $\beta = 0.8$ and the 05:00–10:30 time slot, and the resulting routes are sorted by preference.

Recommended Route Result. The route list is shown on the left-hand side of the panel sorted by parameters that the user selects. As shown in Figure 10, our system recommends that the user takes the R7 subway line and then makes a transfer on the R5 line to the R2 line. The user could obtain detailed information by clicking on the bar of the route.

Display Map. The system can display all routes and highlight the selected route on the map. As shown in Figure 10, the route ranking on the 1st is the highlighted state.

Display Comfort Degree Diagram. The system can display the recommended routes' comfort degree in a 2D histogram. As shown in Figure 11, different comfortable routes have different colors; the red histogram represents the most comfortable route, whereas the purple histogram represents the opposite.

Display Traffic Flow Diagram. The system can display the traffic flow of different types of vehicles during different time periods. As shown in Figure 12, it provides the actual usage volume of three types of vehicle at 11 a.m. on April 28, 2017, which could help us to understand the law of the change of passenger flow.

The database, which stores history records, updates every day with newly collected data. The transfer flow prediction model is trained offline so that users could obtain an instant result when they interact with the system.

8 RELATED WORK

Traffic Passenger Flow Forecast. There are many studies on traffic passenger flow forecasting. A new type of social data source was used to predict subway passenger flow in [22]. By combining smart card fare collection system data and GPS tracing, the system's data for public transportation predicted real-time passenger flow for urban buses in [36]. In [20], the authors used the popular deep learning method to predict passenger flow based on big data. In [37], the authors applied the dynamic model to traffic passenger flow forecasting. In [14, 31], the authors predicted passenger flow based on different cycles.

Urban Data Mining. There is an urban computing community that addresses smart city challenges using urban geography and human mobility data. The work in [10] extracted the features from online user reviews and offline moving behavior to determine high-value residential communities. The work in [26] analyzed human movement using cellular data in Boston to recommend social events. The work in [35] recommended urban POIs using user check-in behavior data.

Multi-Task and Multi-View Learning. MTL aims to learn multiple related and similar tasks simultaneously to improve the generalization performance of each task. Compared with traditional single task learning, MTL has achieved better results by using a shared representation of features [3]. Evgeniou et al. demonstrated that adding extra regularization for MTL can address the problem of joint feature selection for a group of related tasks [1]. The work in [2, 24] proposed joint feature learning methods for MTL. Argyriou et al. generalized the well-known lasso from a single task case to multi-task case [28]. Many tasks are mutually related, however, they are not always strongly related. Recent studies have presented different task relationship structures for clustered structures [15], tree structures [18], and graph structures [5]. A more recent study in [38] proposed a novel MTL framework to solve the critical problem in which the subtasks are spatially correlated.

Route Recommendations. Our work is relevant to route recommendations. In public transportation systems, transportation services are time-dependent and spatially constrained [8, 34]. To model the decision process of choosing routes, researchers have proposed considering more factors, including number of transfer passengers, walking distance or time for passengers, and

crowdedness [16]. The recent work in [27, 29] proposed methods to minimize travel time in public transportation networks. More studies have been conducted to incorporate the capacity constraints of transportation networks into route recommendations; for instance, frequency-based networks [32] and schedule-based networks [13]. The work in [21] developed a strategy called a hyperpath, which is a set of elementary paths that connect an origin to a destination at a given time to recommend routes. Particularly, given an origin and destination, multiple candidate paths could be generated for a user. Then, route recommendations were managed by hyperpath-based algorithms in [4, 6, 23].

9 DISCUSSION

We implemented a model for multi-task transfer passenger flow forecasting based on different spatiotemporal features, and an optimal route recommendation system based on the prediction results in this article.

For the MTPFP model, we extracted a variety of discriminative features from pre-treated multi-source data first, which included subway station features and historical transfer passenger flow features. Then, we defined the basic objective function and adopted several strategies to optimize it to make the algorithm more robust. Next, we designed a specific machine learning algorithm to train the model using the previously extracted features to minimize the target function. After several rounds of training, we obtained prediction results and evaluated them. Using these methods together, we obtained a good result. We determined the reason that our model's results were excellent through analysis. First, spatial features and temporal features provided varying degrees of support for the accuracy of the prediction results, such as various associations between different stations, different moments, and geographic features. Second, combining the MTL strategy and regularization optimization ensured an intrinsic correlation between the prediction results of different stations, and the efficiency and accuracy of the algorithm. All of these helped the model to improve the accuracy of the results.

For the CARR model, we first generated candidate paths, second, sorted the candidate path sets by crowdedness and time cost, and finally, obtained results by calculating the priority functions of different routes. By introducing a comfort index, our model can recommend a more humane route that is superior to the existing widely used map service, and it is also able to adjust results based on different time periods dynamically.

In conclusion, the above experimental results also confirmed the advantages and effectiveness of our approach and models.

10 CONCLUSION AND FUTURE WORK

In this article, we presented a novel framework called MTPFP to first predict the transfer volumes of each transfer station, which was achieved using an MTL-based regression approach. Based on this result, we provided a novel route recommendation model (CARR) which considers the balance of route travel time and crowdedness. Our work provides a new perspective on the quantitative description of urban passenger flows, and the established prototype system can help users to select a comfortable travel route that can avoid crowds. In comparison with conventional methods, our work demonstrates the advantage of the effectiveness of using real data.

Despite the promising performance of the proposed method, there is still room for improvement. First, we only used subway station data in the model for prediction; however, there are still many other traffic data sources that can be used, such as bus and taxi, which may enrich the present model. Second, the accuracy of the multi-task prediction algorithm still needs to be improved; for example, we can add some new regular terms and constraint conditions or remove some of the poor features, and then extract the new and more effective features, or even adopt better machine

learning methods to make the model more accurate. Third, the route generating algorithm of our CARR model may be inefficient when the number of stations increases; thus, we should design new algorithms to solve this problem. Future work will focus on these issues.

REFERENCES

- [1] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2007. Multi-task feature learning. *Advances in Neural Information Processing Systems* 19 (2007), 41.
- [2] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2008. Convex multi-task feature learning. *Machine Learning* 73, 3 (2008), 243–272.
- [3] Rich Caruana. 1998. Multitask learning. *Learning to Learn*. Springer, 95–133.
- [4] I. Chabini and Shan Lan. 2002. Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks. *IEEE Transactions on Intelligent Transportation Systems* 3, 1 (2002), 60–74.
- [5] Xi Chen, Qihang Lin, Seyoung Kim, Jaime G. Carbonell, and Eric P. Xing. 2010. Smoothing proximal gradient method for general structured sparse regression. *Annals of Applied Statistics* 6, 2012 (2010), 719–752.
- [6] E. W. Dijkstra. 1959. A note on two problems in connection with graphs. *Numerische Mathematics* 1, 1 (1959), 269–271.
- [7] Bowen Du, Chuanren Liu, Wenjun Zhou, Zhenshan Hou, and Hui Xiong. 2016. Catch me if you can: Detecting pick-pocket suspects from large-scale transit record. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*, 87–96.
- [8] Steven Farber and Liwei Fu. 2017. Dynamic public transit accessibility using travel time cubes: Comparing the effects of infrastructure (dis) investments over time. *Computers, Environment and Urban Systems* 62 (2017), 30–40.
- [9] Robert J. Franzese, Jude C. Hays, and Lena M. Schaffer. 2010. Spatial, temporal, and spatiotemporal autoregressive probit models of binary outcomes: Estimation, interpretation, and presentation. APSA 2010 Annual Meeting Paper.
- [10] Yanjie Fu, Yong Ge, Yu Zheng, Zijun Yao, Yanchi Liu, Hui Xiong, and Jing Yuan. 2014. Sparse real estate ranking with online user reviews and offline moving behaviors. In *Proceedings of ICDM*. 120–129.
- [11] Jun Gao and Peter Revesz. 2006. Voting prediction using new spatiotemporal interpolation methods. In *Proceedings of the 2006 International Conference on Digital Government Research*. Digital Government Society of North America, 293–300.
- [12] Zhan Guo and Nigel Wilson. 2007. Modeling effects of transit system transfers on travel behavior: Case of commuter rail and subway in Downtown Boston, Massachusetts. *Transportation Research Record: Journal of the Transportation Research Board* 2006 (2007), 11–20.
- [13] Younes Hamdouch and Siriphong Lawphongpanich. 2008. Schedule-based transit assignment model with travel strategies and capacity constraints. *Transportation Research Part B: Methodological* 42, 7–8 (2008), 663–684.
- [14] Zhongsheng Hou and Xingyi Li. 2016. Repeatability and similarity of freeway traffic flow and long-term prediction under big data. *IEEE Transactions on Intelligent Transportation Systems* 17, 6 (2016), 1786–1796.
- [15] Laurent Jacob, Francis Bach, and Jean Philippe Vert. 2008. Clustered multi-task learning: A convex formulation. *Advances in Neural Information Processing Systems* (2008), 745–752.
- [16] Alireza Khani, Mark Hickman, and Hyunsoo Noh. 2015. Trip-based path algorithms using the transit network hierarchy. *Networks and Spatial Economics* 15, 3 (2015), 635–653.
- [17] Milan Kilibarda, Melita Perčec Tadić, Tomislav Hengl, Jelena Luković, and Branislav Bajat. 2015. Global geographic and feature space coverage of temperature data in the context of spatio-temporal interpolation. *Spatial Statistics* 14 (2015), 22–38.
- [18] Seyoung Kim. 2010. Tree-guided group Lasso for multi-task regression with structured sparsity. In *International Conference on Machine Learning*. 543–550.
- [19] Yanchi Liu, Chuanren Liu, Nicholas Jing Yuan, Liang Duan, Yanjie Fu, Hui Xiong, Songhua Xu, and Junjie Wu. 2014. Exploiting heterogeneous human mobility patterns for intelligent bus routing. In *Proceedings of ICDM*. IEEE, 360–369.
- [20] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. 2015. Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems* 16, 2 (2015), 865–873.
- [21] S. Nguyen and S. Pallottino. 1988. *Traffic Assignment for Large Scale Transit Networks*. Springer, Berlin, 176–186.
- [22] Ming Ni, Qing He, and Jing Gao. 2017. Forecasting the subway passenger flow under event occurrences with social media. *IEEE Transactions on Intelligent Transportation Systems* 18, 6 (2017), 1623–1632.
- [23] H. Noh, M. Hickman, and A. Khani. 2012. Logit-based congested transit assignment using hyperpaths on a scheduled transit network. In *Proceedings of the 4th International Symposium on Dynamic Traffic Assignment*.
- [24] Guillaume Obozinski, Ben Taskar, and Michael I. Jordan. 2010. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing* 20, 2 (2010), 231–252.
- [25] R. Kelley Pace, Ronald Barry, John M. Clapp, and Mauricio Rodriguez. 1998. Spatiotemporal autoregressive models of neighborhood effects. *The Journal of Real Estate Finance and Economics* 17, 1 (1998), 15–33.

- [26] Daniele Quercia, Neal Lathia, Francesco Calabrese, Giusy Di Lorenzo, and Jon Crowcroft. 2010. Recommending social events from mobile phone location data. In *Proceedings of ICDM*. 971–976.
- [27] Nguyen Sang, Stefano Pallottino, and Michel Gendreau. 1998. Implicit enumeration of hyperpaths in a logit model for transit networks. *Transportation Science* 32, 1 (1998), 54–64.
- [28] B. Schölkopf, J. Platt, and T. Hofmann. 2006. Multi-task feature learning. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*. 41–48.
- [29] Heinz Spiess and Michael Florian. 1989. Optimal strategies: A new assignment model for transit networks. *Transportation Research Part B: Methodological* 23, 2 (1989), 83–102.
- [30] Balaji Vasan Srinivasan, Ramani Duraiswami, and Raghu Murtugudde. 2010. Efficient kriging for real-time spatio-temporal interpolation. In *Proceedings of the 20th Conference on Probability and Statistics in the Atmospheric Sciences*. American Meteorological Society, 228–235.
- [31] Huachun Tan, Yuankai Wu, Bin Shen, Peter J. Jin, and Bin Ran. 2016. Short-term traffic prediction based on dynamic tensor completion. *IEEE Transactions on Intelligent Transportation Systems* 17, 8 (2016), 2123–2133.
- [32] Fitsum Teklu. 2008. A stochastic process approach for frequency-based transit assignment with strict capacity constraints. *Networks and Spatial Economics* 8, 2 (2008), 225–240.
- [33] Yilun Wang, Yu Zheng, and Yexiang Xue. 2014. Travel time estimation of a path using sparse trajectories. In *Proceedings of KDD*. ACM, 25–34.
- [34] Yi-Hwa Wu and Harvey J. Miller. 2001. Computational tools for measuring space-time accessibility within dynamic flow transportation networks. *Journal of Transportation and Statistics* 4, 2/3 (2001), 1–14.
- [35] Josh Jia-Ching Ying, Wen-Ning Kuo, Vincent S. Tseng, and Eric Hsueh-Chan Lu. 2014. Mining user check-in behavior with a random walk for urban point-of-interest recommendations. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 3 (2014), 40.
- [36] Jun Zhang, Dayong Shen, Lai Tu, Fan Zhang, Chengzhong Xu, Yi Wang, Chen Tian, Xiangyang Li, Benxiong Huang, and Zhengxi Li. 2017. A real-time passenger flow estimation and prediction method for urban bus transit systems. *IEEE Transactions on Intelligent Transportation Systems* 18, 11 (2017), 3168–3178.
- [37] Jing Zhao and Shiliang Sun. 2016. High-order Gaussian process dynamical models for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems* 17, 7 (2016), 2014–2019.
- [38] Liang Zhao, Qian Sun, Jieping Ye, Feng Chen, Chang Tien Lu, and Naren Ramakrishnan. 2015. Multi-task learning for spatio-temporal event forecasting. In *Proceedings of KDD*. 1503–1512.

Received October 2017; revised March 2018; accepted June 2018