Predicting Passenger's Public Transportation Travel Route Using Smart Card Data

Chen Yang[1], Wei Chen[1], Bolong Zheng[2,3], Tieke He[4],
Kai Zheng[1], and Han Su[1](✉)

[1] Big Data Research Center,
University of Electronic Science and Technology of China, Chengdu, China
{chenyang,weichen,kaizheng,hansu}@uestc.edu.cn, suan.sue@gmail.com

[2] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China

[3] Department of Computer Science, Aalborg University, Aalborg, Denmark
zblchris@gmail.com

[4] State Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing, China
hetieke@gmail.com

Abstract. Transit prediction is a important task for public transport institutions and urban planners to provide better transit scheduling and urban planning. In recent years, there are a lot of research on traffic prediction, but the existing works focus predicting the monolithic traffic trend, and few works focus on passenger's public transportation travel route. In this paper, we study the passenger's travel route and duration prediction. We propose a prediction model based on LSTM neural network to predict passenger's travel route and duration. Specifically, we leverage multimodal embedding to extract passenger's features which are highly related to passenger's travel route and then use a LSTM-based model to improve the prediction accuracy. To verify the effectiveness of our model, we conduct extensive experiments using a real dataset which is collected from Brisbane in Australia for four months. The experimental results show that the accuracy of our model is better than baseline models.

Keywords: Transit prediction · Multimodal embedding · Smart card

1 Introduction

With the growing awareness of environmental protection, people are more and more like to take public transport. Public transportation access and corridors are natural focal points for economic and social activities. These activities help create strong neighborhood centres that are economically stable, safe, and productive. A number of studies have shown that the ability to travel conveniently in an area without a car is an important component of a community's livability. Public

Y. Cai et al. (Eds.): APWeb-WAIM 2018, LNCS 10988, pp. 199–213, 2018.
https://doi.org/10.1007/978-3-319-96893-3_15

transportation provides opportunity, access, choice, and freedom, all of which contribute to an improved quality of life.

Under the theme of intelligent city and intelligent transportation, more and more people use smart card to take buses. Therefore, large scale passengers travel data can be collected. The smart card system can automatically and efficiently record passenger's travel routes and transactions without any additional equipment [2]. It is very important for the research and development of urban computing [11]. Through the processing of the passenger's historical data, we can predict the passenger's public transportation travel route and duration. On the one hand, according to the prediction result of passenger's public transportation travel route and duration, we can predict traffic peak time of city and formulate corresponding bus scheduling policy to alleviate the current bus scheduling imbalance and reduce passenger's waiting time. Further more, by analysing passenger's public transportation travel route and duration, we can depict the connection between the urban areas and plan out more reasonable bus routes, making people transfer less times.

First, we propose two baseline prediction models, which are based on Bayesian and Random Forest(RF) respectively. One model uses naive Bayes which makes conditional independence assumption and another model uses RF which treat information gain ratio as the criteria of attribute division. According to the historical dataset, models predict the passenger's current travel route. However, the prediction models based on Bayesian and RF have the following shortcomings: (1) the models do not take into account the impact of passenger's travel route at different time periods, and they only make an independent prediction of passenger's travel route at a certain time period. (2) the models do not predict the passenger's travel duration.

To address these challenges, this paper propose Long Short-Term Memory (LSTM)-based prediction model that enables accurately predict the passenger's travel route and duration from the passenger's historical dataset. The model is based on the LSTM [7], which maps the passenger's travel duration to different time periods. Through the interaction of each neuron, the model predicts the passenger's travel route at each time period. Then the passenger's travel route and duration are predicted under the given condition of features.

Built upon the LSTM, the prediction model uses multimodal embedding to achieve higher prediction accuracy. Before the training model, the features and labels are preprocessed by multimodal embedding [17]. The multimodal embedding learner maps all the passenger, time, week, stop and route units into the same space with their correlation preserved. If two units are highly correlated, then the distance is very close between their distributed representations of vectors. The multimodal embedding not only allows us to capture the similarity between subtle semantic units, but also provides us with background information, which reveals the relationship among passenger, time, week, stop and route.

In summary, we make the following major contributions in this paper:

1. We propose two kinds of baseline models, including Bayesian-based prediction model and RF-based prediction model. But there are a lot of defects in two models. So we design the prediction model based on LSTM. The model can not only predict the passenger's travel route, but also predict the passenger's travel duration. Compared with the baseline algorithm, the model has higher accuracy.
2. We employs a multimodal embedding learner that jointly maps the passenger, time, week, stop and route into a latent space with their correlation preserved. Such multimodal embedding not only make us to capture the subtle semantics of travel records, but also serve as background knowledge to extract features for travel records.
3. We conduct massive experiments using smart card dataset by 83515 travel records over a period of four months. The experimental results show that the proposed algorithm is very effective and outperforms baselines significantly. It can predict the passenger's travel route and duration very accurately.

The remainder of the paper is organised as follows: Sect. 2 introduces preliminary concepts and the work-flow of the proposed model. Section 3 introduces the baseline models. Our proposed prediction model is presented in Sect. 4. The experimental results are presents in Sect. 5, followed by a brief review of related work in Sect. 6. Section 7 concludes the paper.

2 Problem Statement

In this section, we introduce preliminary concepts and formally define the problem. We summarize the major notations used in the rest of the paper in Table 1.

Table 1. Summary of notations

Notation	Definition
r	A bus route
u	A passenger
t	The tth time of the day
s	The sth stop in all station
w	Denote weekday or weekend
w_i	An input vector in multimodal embedding
w_o	An output vector in multimodal embedding
S	Feature set in dataset
x^t	The input vector of the tth neuron in the prediction model
y^t	The label vector of the tth neuron in the prediction model
$\tilde{y}^{(t)}$	The output vector of the tth neuron in the prediction model

2.1 Preliminary Concepts

Definition 1 *TIME PERIOD. Time period is the discrete number that identifies the time period in a day. We divide each k minute into a time period from 0:00 to 24:00 in a day.*

Definition 2 *STOP. A stop s is a fixed location in the space where a passenger u get on or get off the bus in public transportation.*

Definition 3 *ROUTE. A route r is a bus route, which is consisted of a set stops, i.e., $r = [s_1, s_2, \ldots, s_m]$.*

Definition 4 *TRAVEL ROUTE. Travel route is a route r which a passenger takes at time period t at stop s.*

Definition 5 *TRAVEL DURATION. Travel duration is the time duration which a passenger spends on a route r. When a passenger u gets on the bus at t_1 and gets off the bus at t_i, his travel duration is about $(t_i - t_1) \cdot k$.*

2.2 Problem Description

Given a passenger u, his location s and the current time t, predict the public transportation travel route which he will take and how long he will stay on the public transportation travel route.

2.3 System Overview

In this section, we present the work-flow of generating prediction result.

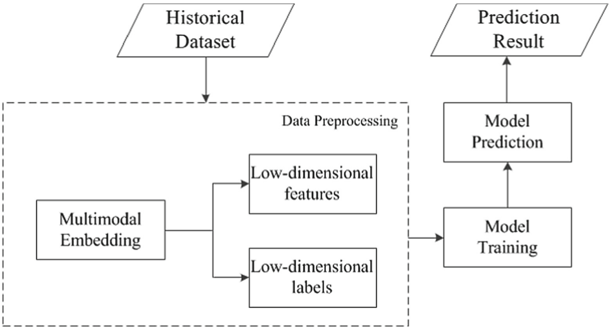


Fig. 1. System overview of model

Figure 1. Shows the system overview of prediction model. The input of the model is historical dataset, and the output is prediction result. The framework contains three main component steps: data preprocessing, model training, and

model prediction. The data preprocessing uses the embedding learner including skip-gram and embedding that maps the passenger, time, week, stop and route units into the same low-dimensional space using extensive data from historical dataset. Then the model is trained and predicted based on the preprocessed data, and finally the prediction result is generated.

3 Baseline Models

We propose two baseline models. The first prediction model is based on Bayesian which makes conditional independence assumption and the second prediction model is based on RF which treat information gain ratio as the criteria of attribute division.

3.1 Prediction Model Based on Bayesian

It is very difficult to directly predict the passenger's travel route under given conditions. In order to solve the posterior probability problem, we adopt the Naive Bayes decision algorithm. The algorithm makes a strong hypothesis-attribute conditional independence assumption. It transforms the posterior probability (which is very difficult to be solved) to the prior probability(which is easy to solve).

We define the prediction model to predict passenger's travel route under certain conditions. Different models are generated for different passengers using Eq. (1):

$$r = \arg\max_{r} p(r|t, s, w) \tag{1}$$

where t is travel time, s is a stop by stopID, and w is a tag indicating whether it is a weekday or weekend(if w is weekend, then w=0, otherwise w=1), $p(r|t, s, w)$ is the probability of the passenger choosing to take route r under the conditions of t, s and w.

It is very natural to calculate posterior probability $p(r|t, s, w)$ with Naive Bayesian classification algorithm. The formula is shown as following:

$$p(r|t, s, w) = \frac{p(t, s, w|r)p(r)}{p(t, s, w)} \tag{2}$$

where $p(t, s, w|r)$ is conditional probability(or likelihood). Because the Naive Bayesian model assumes that the conditions are independent and identically distributed, the likelihood can be evaluated by Eq. (3). $p(r)$ is the prior probability calculated by Eq. (4). $p(t, s, w)$ is the probability of known condition and the value is constant C.

$$p(t, s, w|r) = P(t|r)p(s|r)p(w|r) = \frac{|D_{t,r}|}{|D_r|}\frac{|D_{s,r}|}{|D_r|}\frac{|D_{w,r}|}{|D_r|} \tag{3}$$

$$p(r) = \frac{|D_r|}{|D|} \tag{4}$$

D is the number of training dataset samples; $|D_r|$ is the number of records in D that the passenger takes route r; $|D_{t,r}|$ is the number of records in D that the passenger takes route r at the time period t; $|D_{s,r}|$ is the number of records in D that the passenger takes route r in the stop s; $|D_{w,r}|$ is the number of records in D that the passenger takes route r under the condition of w. After calculating likelihood probability by Eq. (3) and the prior probability by Eq. (4), we can calculate the posterior probability of passenger choosing to take route r under the conditions of t, s, and w by Eq. (5):

$$p(r|t,s,w) = p(t|r)p(s|r)p(w|r)p(r) = \frac{|D_{t,r}||D_{s,r}||D_{w,r}|}{|D_r||D_r||D|C} \tag{5}$$

Then, we enumerate all routes passing stop s and evaluate all the probabilities of a passenger to travel each route. At last, we choose the route with the greatest probability as the result.

3.2 Prediction Model Based on RF

Random Forest(RF) adopts the idea of ensemble learning. It takes decision tree as a base learner, then votes the classification results of all the decision trees, and finally selects the classification result with the largest number of votes as the result of prediction. RF makes use of sample disturbance and attribute disturbance to make the prediction model to achieve high generalization ability.

For RF-based prediction model, we select the features, i.e., passenger, stop, time, week, and use the information gain ratio as criterion. The decision tree is then constructed by m samples of bootstrap sampling and selection of a feature. In this way, we construct M decision trees to form a random forest. Finally, the relative majority voting method is adopted to select the final forecast result as shown in Eq. (6):

$$H(x) = y_{\arg\max_{j} \sum_{i=1}^{M} h_i^j(x)} \tag{6}$$

where M is number of decision trees in random forest, y_j is sample label and h_i is a decision tree in random forest. $h_i^j(x)$ is the output of h_i on y_j(0 or 1).

4 LSTM-based Prediction Model

In the baseline prediction models, passenger's travel route are regarded as discrete to predict, that is, the travel route is predicted respectively between time periods t and $t+1$. In reality, the passenger's travel route at the time period t may have an significant impact on the travel route at the time period $t+1$. Therefore, the prediction accuracy can be improved by considering the impact of passenger's travel route at different time periods. At the same time, the baseline prediction models do not consider getting off time and can not predict the passenger's travel duration.

The Recurrent Neural Network(RNN) model solves the above problems. The neurons in the hidden layer connect each other, and the state of neurons at $t + 1$ is affected by the state of neurons at t, thus RNN can consider the influence of passenger's travel route at different time periods. Meanwhile, the RNN can make the information persistent transfer so as to conduct the serialized prediction. Thus, The RNN model not only predicts the passenger's travel route at the time period t, but also can predicts the travel duration.

Thus, we propose to maximize the probability of correct choice given the passenger's features using the following formulation:

$$\theta^{*} = \operatorname{argmax} \sum_{(X,R)} \log p(R|X;\theta) \tag{7}$$

where X is the input feature, R is the passenger's travel route and θ denotes user-defined parameters of our models. It is common to apply the chain rule to model the joint probability over $R^{(1)}, \ldots, R^{(T)}$, where T is number of hidden layer neurons. $logp(R|X)$ is measured as following:

$$\log p(R|X) = \sum_{t=1}^{T} \log p(R^{(t)}|X, R^{(1)}, \ldots, R^{(t-1)}) \tag{8}$$

We can optimize the sum of the probabilities as described in (8) by using stochastic gradient descent(SGD).

We model $p(R^{(t)}|X, R^{(1)}, \ldots, R^{(t-1)})$ using RNN, where the state of t is determined by the state of $t - 1$ and the input of t. The current state is updated by using nonlinear function f:

$$h^{(t)} = f(h^{(t-1)}, x^{(t)}) \tag{9}$$

4.1 LSTM Model

In the training process of most neural networks, there are problems of gradient vanishing and explosion [7]. In order to solve these challenges, RNN evolves the particular form,called LSTM [7]. LSTM is widely used in natural language processing [1,13], picture and sound capture [3] and sequence prediction [4], and has achieved great success.

The key to LSTM is the cell state, as shown in Fig. 2. The horizontal line running through the top is the cell state. The cell state runs directly throughout the LSTM, and it allow the state of all neurons to be easily transmitted across the entire neural network through a small number of linear operations. The core components of the LSTM structure are the forget gate, the input gate and the output gate. The blue box means forget gate, the green box means input gate and the red box means output gate. The forget gate determines what information is discarded from the cell state. The input gate determines which information is stored in the cell state. The output gate determines which information will be output. The gate value is the number between 0 and 1. If the gate value is 1, all the information will be reserved. If the gate value is 0, all the information will be discarded. The output h at time $t-1$ and the input x at time t together determine the output h at time t through three gates. Cell state c at time $t-1$ is altered by forget gate. Output h at time t is calculated by softmax function, and finally the prediction results are obtained.

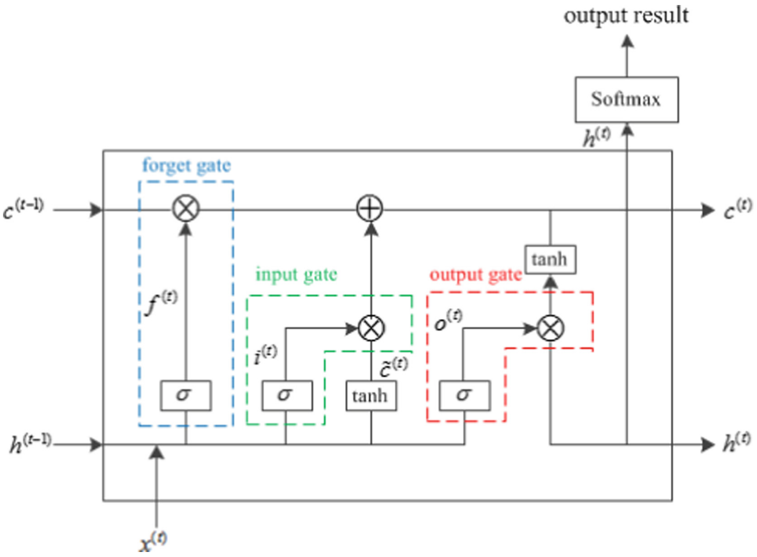


Fig. 2. LSTM cell structure (Color figure online)

The definition of gates and cell update and output are as follows:

$$\begin{aligned}
i^{(t)} &= \sigma(W_{ix}x^{(t)} + W_{ih}h^{(t-1)} + b_i)\\
f^{(t)} &= \sigma(W_{fx}x^{(t)} + W_{fh}h^{(t-1)} + b_f)\\
\tilde{c}^{(t)} &= \tanh(W_{ch}h^{(t-1)} + W_{cx}x^{(t)} + b_c)\\
c^{(t)} &= f^{(t)} \otimes c^{(t-1)} + i^{(t)} \otimes \tilde{c}^{(t)}\\
o^{(t)} &= \sigma(W_{ox}x^{(t)} + W_{oh}h^{(t-1)} + b_o)\\
h^{(t)} &= o^{(t)} \otimes \tanh(c^{(t)})\\
\tilde{y}^{(t)} &= soft\max(h^{(t)})
\end{aligned} \tag{10}$$

where $\otimes$ represents the product with a gate value, W is respectively input weights, output weights, and forget weights, b is respectively the input bias, output bias and forget bias in LSTM networks, $\sigma(\cdot)$ nonlinear sigmoid function and $tanh(\cdot)$ is hyperbolic tangent function. The last equation $\tilde{y}$ is a probability distribution over taking all routes by $h^{(t)}$ feed to a softmax. In the LSTM, multiplicative gates make it possible to deal well with exploding and vanishing gradients.

4.2 Multimodal Embedding

In LSTM-based prediction model, if one-hot encoded passenger, time, week and stop are used as input vector directly, the feature vector will be particularly discrete and the weight matrix will be sparse, which is unfriendly to the training of neural network. At the same time, the dimension of matrix and feature vector is very large, which will cause extremely long training time and prediction time. So we transform one-hot into distributed representation. We employ the method of multimodal embedding [17] which maps all the passenger, time, week, stop, and route units into the same low-dimensional space with their correlations preserved. If two units often appear together, their similarity is very large and their embedding tend to be close in latent space(for example, passenger A often takes a route at B stop, the distance between A and B is very close in latent space). When passenger, week, stop, and route are all natural and discrete units, we can directly use them as embedded units. However, time is continuous and there is no natural embedding units. To address this problem, we divide every k minute into a time period in a day and consider each time period as a basic time unit.

Our embedding algorithm is inspired by the Skip-gram model [10] that predicts the surrounding context by one unit. Here, we regard each element (passenger, time, week, stop, and route) in a travel record as a unit. Given a travel record d, We calculate the similarity between the two units, defined as

$$s(w_i, w_o) = V_{w_i}^T V_{w_o} \tag{11}$$

where V_{w_i} is the embedding of unit w_i, V_{w_o} is the embedding of unit w_o. We model the likelihood using softmax function as follows:

$$p(w_o|w_i) = \exp(s(w_i, w_o)) / \sum_{w \in d} \exp(s(w, w_i)) \tag{12}$$

where w_i is the training feature and w_o is the target feature. $s(w_i, w_o)$ is the similarity score between w_i and w_o.

For all passengers' records dataset S, the objective of the multimodal embedding is to predict all the units in S. We define the loss function as follows:

$$J_S = -\sum_{d \in S} \sum_{w_i \in d} \sum_{-m \leq j \leq m} \log p(w_{i+j}|w_i) \tag{13}$$

where m is the size of the training context. In order to minimize the above loss function, we use the method of stochastic gradient descent(SGD) and negative sampling [10] to update the weight value. We use Noise Contrastive Estimation(NCE) [5] which makes the training time shorter and improves the accuracy of the representation of feature vector. At each time we randomly select a record d from $S(d \in S)$ and randomly select a unit i from $d(i \in d)$. Then we randomly select K negative units that have the same type with i from S (not appear in d). We define negative samples(NEG) by the following objection function.

$$J_d = -\log \sigma(s(w_i, w_o)) - \sum_{k=1}^{K} \log \sigma(-s(w_k, w_i)) \tag{14}$$

where $\sigma(\cdot)$ is the sigmoid function. The updating rules of variable updates can easily be obtained by the above objective function and using SGD.

4.3 Model Description

LSTM predicts the passenger's travel route using Eq. (8). First, multimodal embedding is used to project features and labels into a same z-dimensional space. Here, using v_u, v_t, v_w, v_s and v_r to represent the features(passenger, time, week, stop) and labels(route). Then we use the mean value $\bar{v} = (v_u + v_t + v_w + v_s)/4$ of the features' distributed representation as the input of the first LSTM cell, and the input of the next LSTM cell in turn are the output of the LSTM cell in the previous time. When the output of a certain time is a special value, which does not represent any route, we stop the prediction. Figure 3 shows the expansion form of the prediction sequence. If the maximum travel time in the dataset is l, the longest length of the LSTM prediction sequence is $T = \lceil l/k \rceil$.

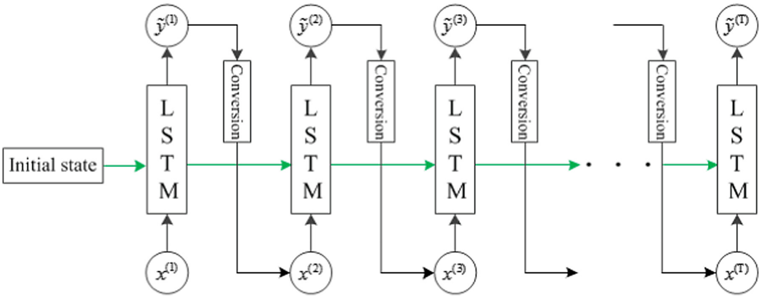


Fig. 3. LSTM-Based prediction model

And as shown in Fig. 3, $x^{(t)}$ is the input vector at the time period $t(1 \leq t \leq T)$. $\tilde{y}^{(t)}$ is the output vector of the model at the time period t, which is represented by a vector, and each element in the vector is the probability of the corresponding prediction result. From the output vector, we choose the route with the maximum probability as the prediction result of the model. $Conversion$ converts the one-hot form of the prediction result into a distributed representation. When $t = 1$, $x^{(t)} = \bar{v}$ and when $t > 1$, $x^{(t)}$ is a distributed representation of prediction results through $Conversion$.

However, a passenger will leave the route at a certain time. To characterize this particular state, we add an additional dimension to the one-hot encoding vector of the route to indicate whether a passenger has left the current route. For example, $(0, \ldots, 0, 1)$ indicate that passenger isn't in a travel route, while $(0, \ldots, 1, \ldots, 0)$ indicate that passenger is in a travel route. The label vector is as follows:

$$y^{(t)} = (r_1, r_2, \ldots, r_{n+1}),\ s.t.\ r_i = 0\ or\ 1 (i = 1, 2, \ldots, n+1) \tag{15}$$

where n is the number of routes.

4.4 Model Training

In the model training phase, we still use the mean of the features' distributed representation as the input vector of the first LSTM cell like the prediction phase. However, unlike the prediction phase, starting from the second LSTM cell, we take the distributed representation corresponding to the true travel route as the input rather than the prediction at last time step. After constructing the model, we use cross entropy as the loss function. The SGD and the back-propagation through time(BPTT) algorithm [12] are used to train the parameters in the LSTM-based model.

5 Experiment

In this section, we present our experimental results to evaluate the performance of proposed prediction model. We conduct the experiments on a computer with Intel Xeon E5-2620 2.10GHz CPU, Titan X GPU, and 128G memory.

5.1 Experimental Setup

DataSet. In our experiments, we use the passenger data in Brisbane, Australia from Translink. We use dataset from January 2013 to April 2013. The information contained in the dataset is shown in Table 2. We choose the 'Inbound' direction in dataset. According to the actual situation, most passengers do not have to travel by more than five hours in a record, so we assume that the maximum travel time is five hours. After removing the noise data. There are 1000 passengers, 3189 stops and 532 routes with a total of 83515 records in dataset. We divide the dataset into a training set and a test set. The training set contains 61646 records, and the test set contains 21869 records.

Table 2. Meaning of each field

Field	Meaning
Smartcard ID	Encrypted unique id of passenger
Direction	Inbound/Outbound
Route	Route number of the bus
Boarding time	Date/Time touch on a card
Alighting time	Date/Time touch off a card
Boarding stop	Boarding stop (ID & Description)
Alighting stop	Alighting stop (ID & Description)

Metrics. To evaluate the performance of all the models, we use the following metrics:

(1) Accuracy. The prediction accuracy is the main factor to measure the performance of the model, $p = N_{true}/N_{total}$, where N_{true} is the number of correctly predicted samples and N_{total} is the number of all the test samples.
(2) Running Time. The time that a model takes to predict the samples is also an effective indicator in measuring the performance of a model. We calculate time from beginning to ending of model predict.

5.2 Performance Evaluation

In this section, we evaluate the performance of the proposed model by conducting both objective and subjective experiments. In Effectiveness Evaluation and Efficiency Evaluation, we set the length of time period $k = 15$ min.

Effectiveness Evaluation. We use the same training set and test set to get the effectiveness of the baseline models and the LSTM-based model we have proposed. We use different number of samples (30 days, 60 days and 90 days) as training sets. The prediction accuracy of different models is shown in Fig. 4. It shows the relationship between the scale of training set and the prediction accuracies of the Bayesian-based, RF-based and LSTM-based respectively. The accuraies in corresponding partitions are referred as a_1, a_2, a_3. From Fig. 4, we have the following two main observations: (1) both a_1 and a_2 are much smaller than a_3, it is because that with multimodal embedding, distributed representation can represent the correlation between features and the interaction between LSTM cells improves the prediction accuracy; (2) as the number of samples increases, both a_1 and a_2 have small increases, while a_3 has a significant change. The reason is that using more training samples means models can better capture the passenger's diverse lifestyle. And for LSTM-based model, each weight matrix can be trained more robustly.

Efficiency Evaluation. We proceed to report the accumulated prediction time of different methods. Table 3 shows the change of running time with the different sample sizes in prediction. From Table 3, we can discover that both Bayesian-based and RF-based models can complete the forecast task in a very short time, while the LSTM-based model needs to take a few seconds due to the matrix operations between the different time steps.

Effects of Parameter. k is the length of time period and can be used to control the number of LSTM cells in LSTM-based model. As a result, k will affect all models' prediction accuracies, and it also has an impact on the prediction time for LSTM-based model. Figure 5(a) demonstrates the change of prediction time for predicting 30000 samples with the increase of the length of time period. This is because that, for a fixed length of time intend when the passenger is

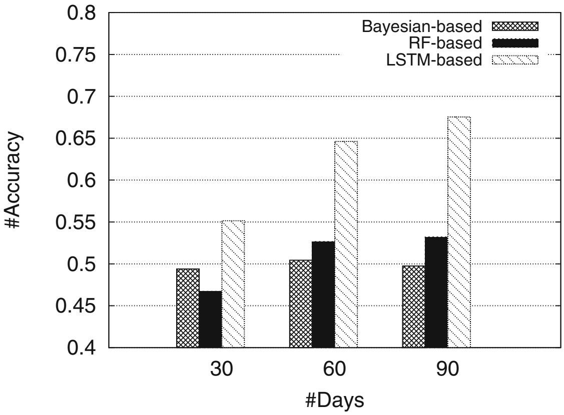


Fig. 4. Prediction accuracy. The total number of samples in different days v.s. accuracy.

Table 3. Prediction time of different models

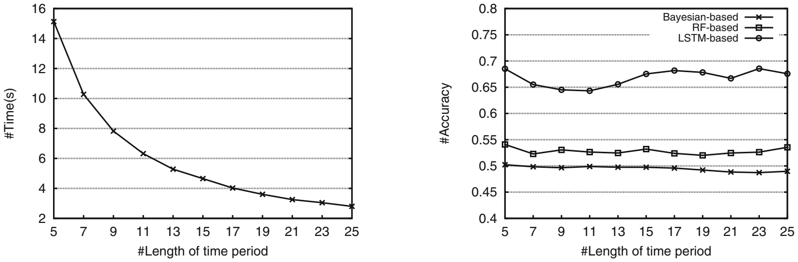
Model	10000	20000	30000	40000	50000
Bayesian-based	0.0030	0.0055	0.0090	0.0130	0.0281
RF-based	0.2552	0.4702	0.7239	0.9430	1.1761
LSTM-based	**2.3155**	**3.0906**	**4.6520**	**6.1690**	**7.7438**

on the route, LSTM-based model only needs a few of time steps to predict the travel duration with a large k. Figure 5(b) demonstrates the relationship between the length of time period and prediction accuracy. From Fig. 5(b), we can observe that the fluctuation of accuracy is very little for Bayes-based and RF-based models, while it is obvious for LSTM-based model. But in general, the fluctuation range is in 0.05. For LSTM-based model, when the length of time period is 23, the prediction accuracy is maximum. But when the length of the time period is increased, the subtle state of the passenger can not be predicted. Therefore, setting k to 15 can better take all aspects of impact into account.

6 Related Work

We study the related work in this section. Most of the previous works are mainly divided into three parts: (1) the study of the urban structure; (2) the recommendation system; (3) passenger's travel destination prediction. However, none of these problems is same with ours.

Urban Structure Discovery. Ma *et al.* [9] put forward the use of spatial clustering and multi criteria analysis to study urban structure. Jiang *et al.* [8] measure spatial and temporal structure of cities by defining space activities with time information.



(a) Prediction time.

(b) Prediction accuracy.

Fig. 5. Efficiency Evaluation

Friends Recommend. Xiao *et al.* [15] proposed an algorithm for measuring the similarity of different users based on the historical track of semantics and then recommend friends to a user. Yu *et al.* [16] proposed a three-step statistical recommendation approach to build a heterogeneous information network.

Destination Prediction. Wang *et al.* [14] proposed a method which insteads of searching similar trajectories in sparse dataset to predict the destination. He *et al.* [6] proposed a model based on kernel density estimation to predict destination by using smart card dataset, and the model achieves a great improvement in prediction accuracy.

7 Conclusions

In this paper we have proposed a LSTM-based model to study passenger's travel route and duration in public transportation using smart card data. As far as we know, we are the first to mention passenger's travel route in urban computing. With the multimodal embedding of the passenger, time, week, stop and route, we extract the features which reserve the correlation in a low-dimension space. We conduct extensive experiments in a real dataset. The experimental results show that the performance of our model is better than the baseline models. In the future, we are interested in extending the method for passenger transfer during a journey, and have a better prediction for the details of individual transit.

References

1. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar, 25–29 October 2014, A Meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1724–1734 (2014)

2. Eagle, N., Pentland, A.S., Lazer, D.: Inferring friendship network structure by using mobile phone data. Proc. Nat. Acad. Sci. U.S.A. **106**(36), 15274–15278 (2009)
3. Gao, L., Guo, Z., Zhang, H., Xu, X., Shen, H.T.: Video captioning with attention-based LSTM and semantic consistency. IEEE Trans. Multimed. **19**(9), 2045–2055 (2017)
4. Graves, A.: Generating sequences with recurrent neural networks. CoRR abs/1308.0850 (2013)
5. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. J. Mach. Learn. Res. **13**, 307–361 (2012)
6. He, L., Trépanier, M.: Estimating the destination of unlinked trips in transit smart card fare data. Transp. Res. Rec. J. Transp. Res. Board **2535**(2535), 97–104 (2015)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
8. Jiang, S., Ferreira Jr. J., González, M.C.: Discovering urban spatial-temporal structure from human activity patterns. In: Proceedings of the ACM SIGKDD International Workshop on Urban Computing, UrbComp@KDD 2012, Beijing, China, 12 August 2012, pp. 95–102 (2012)
9. Ma, X., Liu, C., Wen, H., Wang, Y., Wu, Y.J.: Understanding commuting patterns using transit smart card data. J. Transp. Geogr. **58**, 135–145 (2017)
10. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, Proceedings of a Meeting held 5–8 December 2013, Lake Tahoe, Nevada, United States, pp. 3111–3119 (2013)
11. Paulos, E., Goodman, E.: The familiar stranger: anxiety, comfort, and play in public places. In: Proceedings of the 2004 Conference on Human Factors in Computing Systems, CHI 2004, Vienna, Austria, 24–29 April 2004, pp. 223–230 (2004)
12. Pearlmutter, B.A.: Gradient calculations for dynamic recurrent neural networks: a survey. IEEE Trans. Neural Netw. **6**(5), 1212–1228 (1995)
13. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, Montreal, Quebec, Canada, 8–13 December 2014, pp. 3104–3112 (2014)
14. Wang, L., Yu, Z., Guo, B., Ku, T., Yi, F.: Moving destination prediction using sparse dataset: a mobility gradient descent approach. TKDD **11**(3), 37:1–37:33 (2017)
15. Xiao, X., Zheng, Y., Luo, Q., Xie, X.: Inferring social ties between users with human location history. J. Ambient Intell. Humaniz. Comput. **5**(1), 3–19 (2014)
16. Yu, X., Pan, A., Tang, L.A., Li, Z., Han, J.: Geo-friends recommendation in GPS-based cyber-physical social network. In: International Conference on Advances in Social Networks Analysis and Mining, pp. 361–368 (2011)
17. Zhang, C., Zhang, K., Yuan, Q., Peng, H., Zheng, Y., Hanratty, T., Wang, S., Han, J.: Regions, periods, activities: uncovering urban dynamics via cross-modal representation learning. In: Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, 3–7 April 2017, pp. 361–370 (2017)