TDP: Personalized Taxi Demand Prediction Based on Heterogeneous Graph Embedding

Zhenlong Zhu[1], Ruixuan Li[1*], Minghui Shan[2], Yuhua Li[1*], Lu Gao[1], Fei Wang[2], Jixing Xu[2], Xiwu Gu[1]
[1]School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
[2]BizTech Dept., Didi Chuxing, Beijing, China
{zzlong,rxli,idcliyuhua,gaolu_9585,guxiwu}@hust.edu.cn
{Shanminghui,unsungwangfei,xujixing}@didiglobal.com

ABSTRACT

Predicting users' irregular trips in a short term period is one of the crucial tasks in the intelligent transportation system. With the prediction, the taxi requesting services, such as Didi Chuxing in China, can manage the transportation resources to offer better services. There are several different transportation scenes, such as commuting scene and entertainment scene. The origin and the destination of entertainment scene are more unsure than that of commuting scene, so both origin and destination should be predicted. Moreover, users' trips on Didi platform is only a part of their real life, so these transportation data are only few weak samples. To address these challenges, in this paper, we propose Taxi Demand Prediction (TDP) model in challenging entertainment scene based on heterogeneous graph embedding and deep neural predicting network. TDP aims to predict next possible trip edges that have not appeared in historical data for each user in entertainment scene. Experimental results on the real-world dataset show that TDP achieves significant improvements over the state-of-the-art methods.

CCS CONCEPTS

• **Information systems** → **Users and interactive retrieval**; • **Computing methodologies** → *Artificial intelligence*; • **Applied computing** → Computers in other domains.

KEYWORDS

taxi demand prediction; heterogeneous graph embedding; deep neural network

ACM Reference Format:
Zhenlong Zhu, Ruixuan Li, Minghui Shan, Yuhua Li, Lu Gao, Fei Wang, Jixing Xu, Xiwu Gu. 2019. TDP: Personalized Taxi Demand Prediction Based on Heterogeneous Graph Embedding. In *Proceedings of the 42nd Int'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19), July 21–25, 2019, Paris, France.* ACM, NY, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331368

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331368

1 INTRODUCTION

Traffic is one of the basic and essential part of millions of people's daily life. Rapid development of taxi requesting services, such as Didi Chuxing in China, enable us to collect large-scale transportation data. Through analyzing these data, we can further offer better services and constitute a smarter city. One of the crucial intelligent services is predicting possible trips in a short-term period for a given user, which is called personalized taxi demand prediction.

According to the different scenes of transportation, we can divide taxi demands into several subclasses, for example, the commuting demands and entertainment demands. There are several challenges in this work. First, since unsure origins and destinations are common in the entertainment scene, we should predict those possible paths that haven not appeared in the historical records of the given user. Second, the trip records on Didi platform are only a small part of users' real transportation, which is a weak sampling on the real dataset. Last, the trip records are imbalance in different scenes. For example, users have a large number of records of commuting scene but only a small number of entertainment records. Therefore, it is difficult to represent a user with a proper vector. Existing predicting methods [3, 6, 9] in transportation systems focus more on the destination and the traffic time, which are less helpful for our task.

In this paper, we focus on the entertainment scene, and propose the Taxi Demand Prediction (TDP) model to figure out how to filter the most possible directive paths for each user. In TDP, based on heterogeneous graph embedding (HGE) method, we first embed the regions and users into a latent space to extract important features. Then, we select the candidate origin set and candidate destination set for each given user according to data analysis. Finally, the given user's most possible taxi demands on these two candidate sets can be predicted based on his/her portrait information, which is obtained from a deep neural predicting network.

In summary, the main contributions of this paper are as follows. First, TDP is the first work proposed to predict both origin and destination of next possible trip. Second, in order to solve the class imbalance problem, we embed the regions and users based on a novel heterogeneous graph embedding (HGE) model. Finally, multiple experiments are conducted on a real-world transportation dataset from Didi platform to evaluate the performance of TDP. The results show that TDP is excellent on personalized taxi demand prediction and outperforms many state-of-the-art algorithms on every key step.

*Corresponding Author

2 METHODOLOGY

2.1 Overall Model

We formally define the format of trip records and the problem of personalized taxi demand prediction as follows. It is worth mentioning that a region has different physical meaning when it is regarded as an origin and as a destination. In this paper, we split regions into two states, the origin and the destination.

Definition1. (Trip Records). Let $R = \{r_1, r_2, \cdots, r_n\}$ be the set of trip records in historical data. A trip record formatted as $r_i = (u, t, ls, le, ls_{tag}, le_{tag}, s)$ is defined as a tuple with seven components, where u denotes the user, t denotes the beginning timeslot of the trip, ls denotes the origin, le denotes the destination, ls_{tag} denotes the type of origin, le_{tag} denotes the type of destination and s denotes the corresponding scene of this trip.

Definition2. (Taxi Demand Prediction). Given a user u, an origin set O, a destination set D, our goal is to predict the strong demands that u needs taxi service from the orgin o to the destination d.

The overall architecture of TDP model is illustrated in Figure 1. From the figure, HGE learns the embeddings for regions based on three heterogeneous graphs. And then the origin, destination and time embeddings are fed into our predicting neural network for demand prediction.

2.2 Heterogeneous Graph Embedding

In the entertainment scene, we have to predict both the origin and the destination of a transportation demand. Thus, the first two key points are embedding users and regions into a new common space, and selecting two candidate sets of origins and destinations.

Represent users and regions HGE constructs three heterogeneous graphs to encode both the co-occurrence and time relationships, shown in Figure 1. We divide regions into two states, origin and destination. *LS* (set of origins) and *LE* (set of destinations) are formally the same.

The origin-destination graph (ODG) is a graph in which $V = \{LS, LE\}$, and it describes the average number of times per day for a trip from origin ls_i to destination le_j. **The origin-timeslot graph (OTG)** is a graph in which $V = \{LS, T\}$, and the weight on each edge is the average number of times per day of the corresponding trip. The trip starts in timeslot t_k and starts from region ls_i which appeared in historical data. In particular, we divide a day into 4 timeslots and treat weekends and workdays differently, so $|T|$ is 8. **The destination-timeslot graph (DTG)** is a graph in which $V = \{LE, T\}$, and the weight on each edge is the average number of times per day of the corresponding trip. The trip starts in timeslot t_k and ends at region le_j which appeared in historical data.

For these three heterogeneous graphs, we use matrix factorization mentioned in [5] to obtain embedding vectors for nodes in the network. For example, we can use Eq.(1) to learn the nodes' representation for ODG.

$$\log\left(vol(ODG)D_{row}^{-1}(ODG)D_{col}^{-1}\right) - \log b = XY^T, \tag{1}$$

where $vol\,(ODG) = \sum_{i=1}^{|LS|}\sum_{j=1}^{|LE|} ODG_{ij}$, $D_{row} = diag((ODG)e)$, $D_{col} = diag((ODG)^T e)$, X is the representation for LS and Y is the representation for LE. We transform the above ODG to tar_{od} as the left hand of Eq.(1), and do the same operations for OTG and DTG.

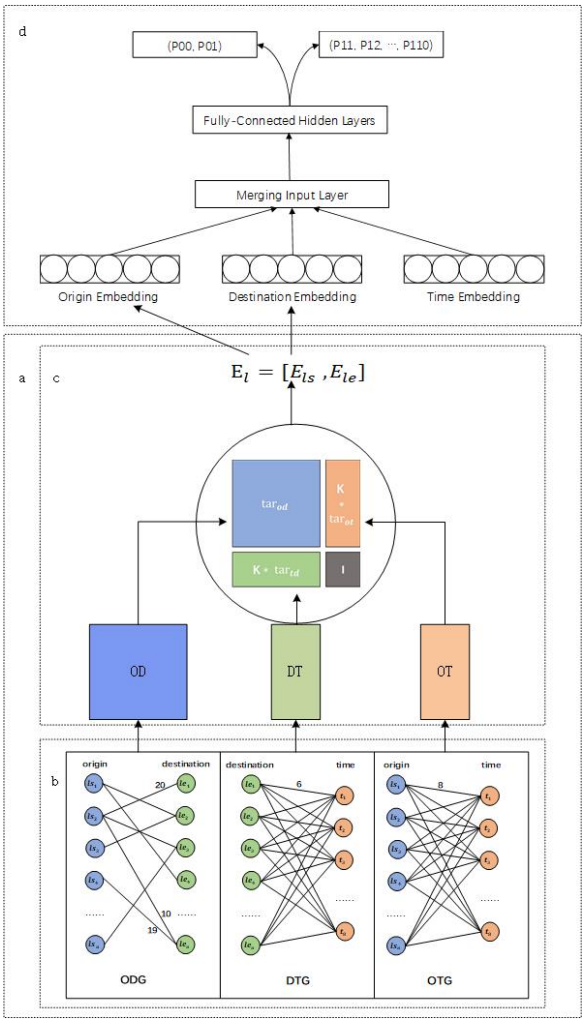

Figure 1: The overall architecture of TDP.

Then we can compose a new whole region matrix M as Figure 1, by using the above three matrix tar_{od}, tar_{ot} and tar_{td}.

Based on above operations, we can get the embedding of origin and destination state respectively for each region l by Eq.(2). K is a hyper-parameter to balance the weights of the three sub-networks. I is an identity matrix, a regularization of the time embedding. The embedding of l can be represented as $E_l = \begin{bmatrix} E_{ls} & E_{le} \end{bmatrix}$. Specifically, the embedding dimensionality of each state for l is 64.

$$\begin{bmatrix} tar_{od} & K * tar_{ot} \\ K * tar_{td} & I \end{bmatrix} = \begin{bmatrix} E_{ls} \\ E_t \end{bmatrix} \begin{bmatrix} E_{le} & E_t \end{bmatrix} \tag{2}$$

According to the type of the regions in dataset, for each user, we can divide regions into three categories: his/her company, residence and entertainment place. So

$$E_u = \left[E_{company}, E_{residence}, E_{entertainment}\right] \tag{3}$$

is a simple idea to concatenate three region vectors of a user to get his/her vector representation.

Obviously, user's entertainment preferences will change over time. According to the life experience, the nearer the entertainment behaviors to the current time, the higher influence they have on the current taxi demands. In order to achieve this, we introduce a time decay function to describe the influence of users' different entertainment behaviors on the current demand. Thus, a user's vector representation of entertainment can be obtained:

$$E_{entertainment} = \frac{1}{n}\sum_{i=1}^{n}\Gamma(\Delta t_i)\, v_i \tag{4}$$

where v_i is the embedding representation of a region i. $\Gamma(\Delta)$ is the time decay function. Circle Kernel [2] is selected as our time decay function through experiments.

Selecting origin and destination candidates Through the analysis of the dataset, we find that there is generally only one region tagged with entertainment in the trip path in entertainment scene. It means users always leave from the origin tagged with entertainment to the destination tagged with other labels, or leave from the origin tagged with other labels to the destination tagged with entertainment. Therefore, for each user, we need to choose a set of entertainment regions and a set of non-entertainment regions as alternative sets of origin and destination.

To construct the set of entertainment regions of a user, we propose two strategies. One is picking two regions tagged entertainment that are most similar with the user's entertainment vector $E_{entertainment}$. Another one is first finding 200 users who are most similar with our target user, then picking two entertainment regions which are most frequently visited by these similar users, using cosine similarity as the measurement method.

Moreover, we choose two non-entertainment regions, tagged with education, resident or company label, to compose the other candidate set. Obviously, these two regions should have the biggest weight in the regions that the target users have visited. The weight can be influenced by the time decay function $\Gamma(\Delta)$.

2.3 Predicting Network

To judge whether all possible region-pairs obtained from these two sets can form a trip and further measure their possibility, we train a neural network with a structure as shown in the Figure 1. The network includes an input layer, multiple fully-connected hidden layers and an output layer. The input of the network is the concatenation of origin embedding, destination embedding and timeslot embedding. We divide a day into 4 timeslots and the timeslots embeddings are randomly initialized and trained just like the network parameters. As mentioned earlier when building the heterogeneous graphs, we divide one day into four periods, and then divided the week into working days and rest days. However, in the process of network training and prediction, we are not sure whether the next trip of users is on a working day or a rest day. Therefore, we do not distinguish them in the network and only maintain the embeddings of four periods in a day. The output layer is composed of two vectors, where $p0 = (p00, p01)$ represents whether the origin and destination of the input can form a possible trip edge. $p00$ is the negative possibility and $p01$ is the positive possibility. $p1 = (p10, p11, ..., p110)$, the value of $p1j$ is the possibility that this input edge may occur on average $2j$ times a day at the input timeslot.

When making taxi demand prediction for a given user with origin l_s and destination l_e, we first caculate the timeslot via Eq(5). where m is the non-entertainment regions in l_s and l_e, t_{m_i} is the timeslot that m appears in the i place.

$$t_m = \frac{1}{n}\sum_{i=1}^{n}\Gamma(\Delta t_i)\, t_{m_i} \tag{5}$$

Then we get $P0$ and $P1$ by DNN and calculate the weight of each input trip (l_s, l_e and t_m) with Eq.(6):

$$W_{l_s l_e} = p_{01}\sum_{j=1}^{10} j * p_{1j} \tag{6}$$

With the above operations, the top5 possible origin-destination transportation demand tuples can be chosen for each user.

3 EXPERIMENTS

3.1 Dataset and Metrics

In this paper, we use a real trip dataset collected from Didi Chuxing. The dataset includes trip records of 20,000 users randomly sampled in a certain Chinese city of two months. We divide regions with fixed acreage instead of specific locations to alleviate data sparsity problem. After analyzing the dataset, we appropriately set the fixed acreage of a region as $0.5km * 0.5km$. Further more, we remove trip edges with a average trip number less than 1 per day, and only reserve these regions with in-degree and out-degree. With above preprocessing, we finally get 2162 valid regions.

In our work, we use Top5 accuracy as the evaluation index. When k=100 in region matrix, TDP model gets the best results for the given trip dataset by experiments. In the following experiments, k is set to 100.

3.2 Performance Comparison

We conduct comprehensive experiments to evaluate the performance of our model on a real dataset, and compare our model with several state-of-the-art baseline models at every key steps, with tuning their parameters.

Comparison on Region Embedding We compare our proposed embedding model HGE with the following state-of-the-art embedding methods. Table 1 shows the performances of above embedding methods and our proposed method HGE.

First, we construct an undirect graph between regions, and embed the regions with DeepWalk [4], Line [8] and AONE [10]. We choose a two-hop sampling for Line and save the 10-order information of the network for AONE. In this case, the Line is called Line1.

Second, we construct two undirect graphs, which describe the relationship of regions to regions and of regions to users. Then, PTE [7] is used for extracting information from it. In this case, the PTE is called PTE1.

Third, with treating time as an attribute of the region, we construct a graph between regions and a graph between region and time. AANE [1] is used to learn the low latitude representation of the region.

Table 1: Performance comparison of embedding methods

Embedding method	DeepWalk	Line1	AONE	PTE1	PTE2	AANE	Line2	HGE
precision	9.81%	11.77%	10.74%	11.35%	11.17%	11.89%	11.91%	12.13%

Table 2: Performance comparison of the strategies for user embedding

User vector	**precision**
Entertainment-Only	9.8%
All-Average	11.5%
Concatenate-Vectors	12.13%

Fourth, we construct a directive bipartite graph between the origin and the destination. Line [8] is used to extract information from it. In this case, the Line is called Line2.

Last, we construct three directive bipartite graphs, which describe the relationship of origin to destination, origin to user and destination to user. Regions are embedded by PTE [7]. In this case, the PTE is called PTE2.

The result of Line2 is superior to DeepWalk, Line1 and AONE, which proved to be meaningful to divide an area into origin and destination these two states. AANE is better than PTE1 and Line1, indicating that it is necessary to consider the time attribute. HGE result is superior to PTE2 and Line2, which validates the correctness of our overall thinking.

Comparison on User Embedding We applied three strategies for user embedding to find the best one. One is directly using the entertainment vector $E_{entertainment}$ as user's embedding, named *Entertainment-Only*. The other one is directly using the weighted average of all regions accessed by the user between different scenes, which is named $All-Average$. The last one is our method $Concatenate-Vectors$. Table 2 shows the performance comparison of these three strategies, and obviously our $Concatenate-Vectors$ method performs better than another two strategies.

The results validates the correctness of our strategy of combining users' vectors in different scenes. The information provided by a single entertainment scene is too little, and the information represented by a single vector in all scenes will lead to information loss due to the unbalanced information in each scene.

Comparison on Predicting Model The traditional method based on statistics also has certain significance for predicting travel demands. It first calculates the average number of times that each possible trip edge appeared in this timeslot period, and use the time decay function to describe the weight of this timeslot period, so as to obtain the transportation possibility of each possible edge. We conduct some experiments on this Statistics-based method with our proposed predicting network, and the results show that our method is better on predicting taxi demands.

4 CONCLUSION

In this paper, we put forward a prediction method TDP of user taxi demands for entertainment scene. In the TDP, we first propose HGE to embed regions, and combine the vectors of users in different scenes to solve the class imbalance problem. Then we select the candidate origin and destination sets for each user. Finally, we feed these two candidate sets to our predicting network model to predict the possibility of the appearance of the origin-destination pairs. In the future work, we will extend this method for more transportation scenes with more data in cities.

Table 3: Performance comparison of path predicting

Edge prediction	**precision**
Statistics-based Method	11.61%
TDP	12.13%

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China under grants 2016YFB0800402 and 2016QY01W0202, National Natural Science Foundation of China under grants U1836204, 61572221, 61433006, U1401258 and 61502185.

REFERENCES

[1] Xiao Huang, Jundong Li, and Xia Hu. 2017. Accelerated attributed network embedding. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 633–641.

[2] Yuqi Li, Weizheng Chen, and Hongfei Yan. 2017. Learning Graph-based Embedding For Time-Aware Product Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2163–2166.

[3] Yaguang Li, Kun Fu, Zheng Wang, Cyrus Shahabi, Jieping Ye, and Yan Liu. 2018. Multi-task representation learning for travel time estimation. In *International Conference on Knowledge Discovery and Data Mining,(KDD)*.

[4] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.

[5] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 2018. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 459–467.

[6] Kohei Tanaka, Yasue Kishino, Tsutomu Terada, and Shojiro Nishio. 2009. A destination prediction method using driving contexts and trajectory for car navigation systems. In *Acm Symposium on Applied Computing*.

[7] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1165–1174.

[8] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1067–1077.

[9] Lingyu Zhang, Tao Hu, Yue Min, Guobin Wu, Junying Zhang, Pengcheng Feng, Pinghua Gong, and Jieping Ye. 2017. A taxi order dispatch model based on combinatorial optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2151–2159.

[10] Ziwei Zhang, Peng Cui, Xiao Wang, Jian Pei, Xuanrong Yao, and Wenwu Zhu. 2018. Arbitrary-order proximity preserved network embedding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2778–2786.