Predicting Origin-Destination Flow via Multi-Perspective Graph Convolutional Network

Hongzhi Shi[1], Quanming Yao[2], Qi Guo[3], Yaguang Li[4*], Lingyu Zhang[5], Jieping Ye[5], Yong Li[1], Yan Liu[6]
[1]Beijing National Research Center for Information Science and Technology (BNRist),
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2]4Paradigm Inc., Hong Kong, China
[3]College of Computer Science, Sichuan University, Chengdu, China
[4]Google Inc., Mountain View, United States
[5]AI Labs, Didi Chuxing, Beijing, China
[6]Department of Computer Science, University of Southern California, Los Angeles, United States

Abstract—**Predicting Origin-Destination (OD) flow is a crucial problem for intelligent transportation. However, it is extremely challenging because of three reasons: first, correlations exist between both origins and destinations; second, the correlations are dynamic across the time; at last, there are multiple correlations from different aspects. To the best of our knowledge, existing models for OD flow prediction cannot tackle all of these three issues simultaneously. We propose Multi-Perspective Graph Convolutional Networks (MPGCN) to capture the complex dependencies. Our proposed model first utilizes long short-term memory (LSTM) network to extract temporal features for each OD pair and then learns the spatial dependency of origins and destinations by a two-dimensional graph convolutional network. Furthermore, we design a dynamic graph together with two static graphs to capture the complicated spatial dependencies and use an average strategy to obtain the final predicted OD flow. We conduct extensive experiments on two large-scale and real-world datasets, which not only demonstrate our design philosophy but also validate the effectiveness of the proposed model.**

Index Terms—**Origin-destination traffic prediction, graph convolutional network, spatial-temporal data analysis.**

I. Introduction

Thanks to the development of ride-hailing software, such as Uber and Didi, many traffic data in the urban area have been widely collected and recorded to support traffic prediction, specifically Origin-Destination (OD) flow prediction. OD flow prediction aims to predict how much volume will be from a specific origin to another specific destination in the next time slot when given some side information. Typical types of side information are history traffic information which records traffic flows in previous time slots, geographic information which describes longitude and latitude of regions, and point of interest (PoI) information which identifies important artificial building inside one region. OD flow prediction is a crucial task in the intelligent transportation area, which benefits a variety of applications such as ride-hailing order dispatching, traffic congestion avoidance, event detection, etc [1], [2]. Many models have been proposed to solve OD flow prediction [1]–[7], the main challenges inside this problem are:

[*] Work done while the author was a PhD student at University of Southern California.

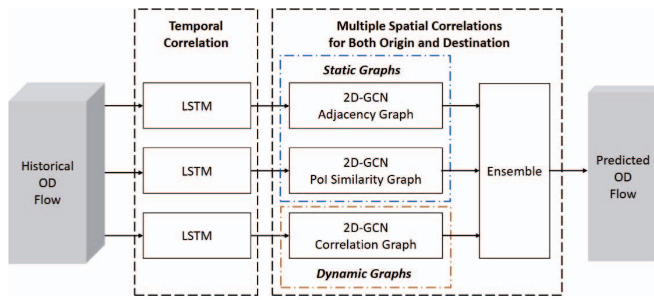


Fig. 1. An overview of the proposed multi-perspective GCN (MPGCN) model. First, MPGCN extracts temporal features from the historical values for each OD pairs, then we use such temporal features to go through multi- layer two-dimensional graph convolutional network to capture both the correlation about origins and destinations, and finally average the outputs of GCN of both static and dynamic graphs for OD pairs prediction.

- **Challenge 1: Two-side Correlations** [5]. For OD flow data, correlations exist in both the origins and the destinations. The OD flows will be dependent when their origins are the same/similar and the destinations are the same/similar. The dependencies between regions are often represented by graphs [3], [4].
- **Challenge 2: Dynamicity** [6]. The correlations between regions are dynamic, which is changing over time. However, the graph constructed by external data sources, such as adjacency graph or PoI similarity graph is usually static, because the external data sources are often not time-varying.
- **Challenge 3: Information Fusion** [4]. Constructed graphs include both static graphs and dynamic graphs, how to effectively and comprehensively fusing and exploiting such multiple aspects of the complex relationships to enhance prediction accuracy for large-scale datasets is difficult.

In this paper, we propose a novel model, i.e., Multi-Perspective Graph Convolutional Network (MPGCN), to simultaneously deal with all the above challenges. First, we conduct graph convolution on both the origin dimension and the destination dimension of OD tensor. Second, we design one dynamic graph for representing the dynamic correlations of regions as origins or destinations calculated by the historical OD flow data. Furthermore, we exploit an average fusion

2375-026X/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDE48307.2020.00178

strategy to capture the complex spatial correlations. Finally, we evaluate our model on two large-scale and real-world datasets collected from Shanghai and Beijing. Experimental results show the effectiveness of the proposed MPGCN over other state-of-the-arts models.

II. Related Work and Background

OD Flow Prediction: OD flow prediction is a crucial problem in intellectual transportation area. To solve this problem, there are traditional models such as Autoregressive Integrated Moving Average model (ARIMA) [8], Support Vector Regression (SVR) [9], factorization-based methods [1], [2], etc; and deep learning based methods [10] usually use RNN (e.g. LSTM or GRU) to capture the long-term dependency of OD pairs to achieve better performance. These models usually cannot successfully capture complex relationships and suffer from low efficiency. Different from them, our model can capture both the long-period dependency and the complicated spatial correlations between different regions for information propagation. There are also abundant works [4], [11] about region-wise flow prediction, which is highly related to OD pair prediction. These works only focus on predicting the value of one of origin and destination instead of both of them, i.e., they predict a value (e.g. in/out flow) for each region, and we need to predict a value for each pair of two regions.

Graph Convolution Network (GCN): Graph convolutional graph (GCN) has achieved the unprecedented success on a series of tasks related to graphs [4], [5], [11]. Given a relationship graph $\mathrm{G} = (\mathrm{V}, \mathrm{E}, \mathbf{A})$, where V is a set of vertices, E is a set of edges and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the connectivity matrix. Then, the graph Laplacian is

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}, \tag{1}$$

where $\mathbf{D}$ is the degree matrix and $\mathbf{I}$ is an identity matrix. GCN generalizes the convolution operation from CNN on graph based on graph Laplacian, which generates a signal from lth layer, i.e., $\mathbf{X}_l$, to $(l+1)$th layer by $\mathbf{X}_{l+1} = \sigma(\sum_{k=0}^{K-1} \alpha_k T_k(\mathbf{L})\mathbf{X}_l)$, where $T_k(\cdot)$ denotes the Chebyshev polynomial [12] of degree k. Aside from 1D graph signal, there are many 2D graph signals in the real world. For example, OD matrix can be regarded as a 2D signal on the graph of origins and the graphs of destinations; user-item rating matrix is a 2D signal on the user network and the item network. Mofnti et al. [5] propose 2DGCN which extends GCN to 2D graph signals. Different from traditional GCN, 2DGCN operates on a matrix that both rows and columns can be regarded as features, i.e., the raw index is corresponding to a node as well as the column index. Specifically, $\mathbf{X} = \{x_{ij}\}$, where x_{ij} means a value associated with the pair of (i-th node in $\mathcal{G}_1$, j-th node in $\mathcal{G}_2$). Thus, the 2DGCN can be defined as $\mathbf{X}_{l+1} = \sigma\left(\sum_{i=0}^{K-1}\sum_{j=0}^{K-1} \alpha_{ij} T_i(\mathbf{L})\mathbf{X}_l T_j(\mathbf{L}^T)\right)$, where $\alpha_{ij} \geq 0$ is the convolution coefficients.

III. Proposed Method

We denote the set of regions as $\mathrm{V} = \{v_1, v_2, \cdots, v_N\}$, where v_i denotes the i-th region. For each time slot, OD flow is represented by a matrix $\mathbf{X}$ where x_{ij} is the number of orders that the taxis go from v_i in this time slot to v_j.

Definition 1 (OD Flow Prediction): Given historical OD $\mathbf{X}_{t-T}, \mathbf{X}_{t-T+1}, \cdots, \mathbf{X}_{t-1}$, the geographic information and PoI information of each regions, we need to predict $\mathbf{X}_t$.

Such OD flows are dynamic across time and contain intrinsic temporal dependency. We exploit LSTM to model such temporal dependency. To model the similarity between regions, we not only use the external data source as side information but also can extract the similarity from the historical OD flows dynamically. Furthermore, there are correlations between origins and destinations requiring joint modeling, so we exploit 2DGCN to model both of them. With all these designs, our model, named Multi-Perspective Graph Convolutional Network, can fit the characteristics of OD prediction problem and achieve advanced performance. To be clearer, the main steps of our method for OD flow prediction are summarized in Algorithm 1.

Algorithm 1 MPGCN: OD flow prediction via multi-perspective GCN.

Require: Historical OD X, Multiple Graphs $\{G\}$;
1: Extract temporal features;
2: Construct Dyn. graph by historical OD flow;
3: Construct Adj. graph by geographic information;
4: Construct PoI. graph by PoI information;
5: **for** graph **in** [Adj., PoI., Dyn.] **do**
6: Train 2DGCN model with graph;
7: Predict the OD flow by 2DGCN with Graph
8: **end for**
9: Integrate the results from the above three models.
10: **return** OD flow in the next time slot.

1) Representing Intra Region Dynamicity by Temporal Features (step 1): In this step, we treat historical values associated with each OD pair as an individual time series and extract temporal features for each OD pair individually. First, we extract five historical values for temporal features construction: the values of one week ago, one day ago and the most recent three time slots. Then, we feed historical OD flow into an LSTM, which can capture both short and long term dependency, and finally obtain the hidden vectors for OD pairs $\mathcal{H}^{(0)}$ as follows, $\mathcal{H}^{(0)} = \mathrm{LSTM}(\mathbf{X}_{t-24\times 7}, \mathbf{X}_{t-24}, \mathbf{X}_{t-3}, \mathbf{X}_{t-2}, \mathbf{X}_{t-1})$.

2) Representing Between-Region Dynamicity by Dynamic Graphs (step 2-4): The key intuition here is that as origins, we use the distributions of how many traffic are from these places to all the destinations as features. First, we average the traffic tensor of the same hours in different days, to form an OD tensor with the temporal dimension length of 24 hours. We denote such tensor as $\mathbf{X}^{(h)}$, where $x_{ijk}^{(h)}$ is the traffic from i-th origin to j-th destination in k-th time slot. To measure the correlation between two origins, we use cosine similarity because (i) the value is limited ranging from 0 to 1; (ii) the larger the cosine value, the more correlated two origins are. Thus the correlation between i_1-th and i_2-th origin is,

$$\mathrm{corr}_t(O_{i_1}, O_{i_2}) = \cos(X^{(h)}[i_1, :, t], X^{(h)}[i_2, :, t]). \tag{2}$$

The correlation between two destination can be calculated similarly. We share the GCN filter parameters across time but use different graphs in different time slots and also different graphs for origins and destinations.

3) Capturing two-side correlation by 2DGCN on Tensor (step 5-8): Motivated by [4], we conduct GCN for both origins and destinations, i.e., we extend GCN into a 2DGCN for such a 2D signal on graphs. Specifically, when passing such a 2DGCN, the calculation is as follows,

$$\mathcal{H}^{(l+1)} = \sigma(\sum_{i=0}^{K-1}\sum_{j=0}^{K-1} \mathcal{H}^{(l)} \times_1 T_i(\mathbf{L_1}) \times_2 T_j(\mathbf{L_2}) \times_3 \mathbf{W_{ij}^{(l)}}), \quad (3)$$

where $\times_n$ means the matrix multiplication on the n-th dimension of the tensor, $\mathbf{L}$ is the Laplacian matrix obtained by Eqn. (1), $\mathbf{W_{ij}^{(l)}}$ is the learnable weight matrix for l-th layer and the degree of (i, j) for GCN, i.e. filter for GCN, $\mathcal{H}^{(l)}$ is the hidden state in the l-th layer whose three dimensions are corrisponding to origins, destinations and feature dimension respectively, and $\sigma(\cdot)$ is the activation function for introducing non-linearity. Such operations on the original 2D graph signal $\mathcal{H}$ have explicit physical meanings: the feature in the position of m-th row and n-th column $\mathcal{H} \times_1 T_i(\mathbf{L_1}) \times_2 T_j(\mathbf{L_2})$ means the average feature that from m-th region's i-hop neighbours to n-th region's j-hop neighbours. We make the temporal features extracted by the temporal correlation module go through multiple layers. On one hand, the information can be propagated more hops by feed through multiple layers; on the other hand, more layers can lead to more non-linearity which is associated with stronger representation ability.

After feeding through several layers of such 2DGCN, we use a linear regression layer for obtaining the predicted traffic matrix, i.e., we aggregate the feature dimension into 1 as following, $\hat{\mathbf{X}} = \mathcal{H}^{(L)} \times_3 \mathbf{W}^{(L)}$, where $\hat{\mathbf{X}} \in \mathbb{R}^{N\times N}$.

4) Information Fusion by Integration of Features (step 9): Multiple relationships exist between regions, such as two regions are close or two regions have PoIs with similar functions. Besides a series of dynamic graphs to capture time-dependent relationship, we construct two static graphs including adjacency graph and PoI similarity graph to capture the time-independent relationships as in [4]. Adjacency graph describes whether two regions are adjacent, while PoI similarity graph represents whether two regions are similar in function reflected by its distribution among different PoI categories. When we use the two static graphs, the graphs for origins and destinations are the same across the time. We separately train three neural networks with three different kinds of graphs, and integrate the prediction results obtained by three models by an average strategy to obtain the best result, which is better than the result obtained by each graph individually.

Since there are some OD pairs usually being 0 across time, so we determine the OD pair which is non-zeros and only take this part into account for calculating the errors. We denoted such a binary matrix by $\mathbf{Y}$ with the same shape of $\hat{\mathbf{X}}$, where the entry $y_{ij} = 1$ if the corresponding pair is usually non-zero (the proportion $\geq 10\%$) and otherwise is 0. By this, we calculate the Mean Square Error (MSE) for the training loss J as $J = \sum_{i,j}(y_{ij}(x_{ij}-\hat{x}_{ij}))^2 / \sum_{i,j} y_{ij}$, which is minimized to train all the parameters of our network. Due to the parameter space of multiple GCNs with different graphs, we tune the hyper-parameters of each GCN individually. For each graph, we first initialize all the parameters randomly and then train the LSTM and GCN jointly, which is optimized by Adam [13]. We average the predicted results from multiple GCNs as the final predicted result.

IV. Evaluation

A. Experiments Setup

1) Datasets: We conduct our experiments on two large-scale ride-hailing datasets collected by Didi Chuxing in two cities in China, i.e., Beijing and Shanghai, from Nov. 8th 2018 to Dec. 30th 2018. We divide each city into small squares of size $3km\times3km$, which generates 17×10 grids in Beijing and 18×9 grids in Shanghai. We have $52,191,284$ ride-hailing records in Beijing and $44,825,098$ ride-hailing records in Shanghai. These eight weeks' data are granular in hours. We split each dataset into three parts: the first four weeks' data for training, the following two weeks' data for validation, and the last two weeks' data for testing.

2) Task Description: As mentioned in Section III, given the historical OD matrix and the relationships between regions represented by graphs, we use different models to predict the OD matrix in the next time slot. In all the following experiments, we set the time granularity as 1 hour. Specifically, given the previous historical values of OD flows, we predict the OD flow in the next time stamp.

3) Evaluation Metric: Following the manner in [1], [4], we use root mean square error for evaluating performance, i.e., $\mathrm{RMSE}(\mathbf{X}, \hat{\mathbf{X}}) = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}(x_{ij}-\hat{x}_{ij})^2 y_{ij} / \sum_{i=1}^{N}\sum_{j=1}^{N} y_{ij}}$, where x_{ij} is the ij-entry of the real flow matrix $\mathbf{X}$, $\hat{x}_{ij}$ is the ij-entry of the predicted flow matrix $\hat{\mathbf{X}}$ and $\hat{y}_{ij}$ is the ij-entry of the indication matrix $\mathbf{Y}$.

4) Compared Methods: The following methods are compared. (i). **History Average (HA)**. It calculates the OD matrix in t-th time slot by averaging the OD matrices of the same time slots in previous weeks. (ii). **Lasso, Ridge**. These two methods use linear regression based on its own historical values with $L1$ regularizer and $L2$ regularizer, respectively. (iii). **LSTM** [10]. It uses the historical values to feed in a LSTM to make the prediction for each time series individually. (iv). **MURAT** [3]. We adapt this method by jointly using graph embedding to extract the features of regions and LSTM to extract temporal features of OD pairs for predicting OD flows. (v). **MGCN** [4]. It conducts the convolution with multiple graphs, which can only capture the relationships from the origin side or the destination side but not the both. (vi). Finally, our model, denoted by **MPGCN**, integrating all results obtained by 2DGCN with adjacency graph, PoI similarity graph and dynamic correlation graph.

5) Experiment Settings: We choose Chebyshev polynomial function [12] for the GCN form and Rectified Linear Units (ReLU) as the activation function after the linear manipulation. In terms of training our model, we exploit Adam [13] for optimizing. To obtain convincing results for performance comparison, we repeat each experiments five times to calculate the mean and standard error of both measurements. All codes are implemented in pytorch and run on a single Tesla P40.

B. Overall Comparison

From the Tab.I, we can observe that our model outperforms all the baselines. Generally, the deep models are better than the traditional models because of the strong expression ability of deep models. For temporal feature extraction, in most cases, we can observe that LSTM is better than linear regression (Lasso and Ridge) because it can better capture the sequential information and introducing non-linear expression ability. Our model can consistently outperform MURAT for both datasets, because our model in an end-to-end way exploit graphs for information propagation. Our method can outperform MGCN, because MGCN can only capture one side dependency, i.e., the relationship between origins or the relationship between destinations but not both of them. Our method can capture the multiple perspectives of OD flow predictions, which can capture both the dependency of origins and destinations, successfully model the dynamic relationships between regions and capture the multi-view relationships. By all of these technical improvements for OD flow prediction, our model outperforms all other method by a large margin.

TABLE I

PERFORMANCE COMPARISON FOR OD PAIR PREDICTION.

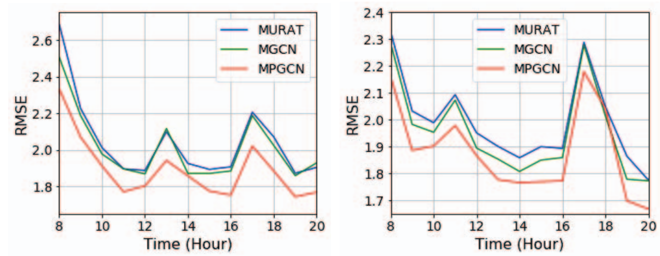
Methods		Beijing RMSE	Shanghai RMSE
Traditional Models	HA	2.145	1.770
	Lasso	2.110 ± 0.025	1.838 ± 0.018
	Ridge	2.093 ± 0.006	1.817 ± 0.001
Deep Models	LSTM	2.008 ± 0.006	1.808 ± 0.003
	MURAT	2.009 ± 0.009	1.830 ± 0.027
	MGCN	1.960 ± 0.004	1.739 ± 0.002
Our Model	MPGCN	$\mathbf{1.869 \pm 0.023}$	$\mathbf{1.678 \pm 0.010}$

C. Results of Different Time Periods

We plot the RMSE of our method and two most representative methods, MURAT and MGCN, in the daytime of workdays and weekends in Fig. 2, because predicting the daytime OD flow with larger values is more meaningful. From Fig. 2(a), we can observe that the profile of weekday of RMSE is basically consistent with the traffic on weekdays and our method consistently outperform the baselines on both weekday and weekend. From Fig. 2(b), we can observe that there are two peaks around $7:00$ and $17:00$ for the RMSE profile on weekend, which is different from the two peaks on weekdays.

V. Conclusion

We dig deep into the spatio-temporal dependence of the problem of OD prediction and designing a GCN-based model



(a) Workday. (b) Weekend.

Fig. 2. Testing RMSE on different time periods in Beijing.

that can successfully capture the dynamic and multiple dependencies for both origins and destinations. We propose multi-perspective graph convolutional model for OD flow prediction, which achieves the best performance against all the state-of-the-art models. We conduct extensive experiments on two large-scale real-world datasets and demonstrate the effectiveness and efficiency of our model.

Acknowledgment

This work was supported in part by The National Key Research and Development Program of China under grant SQ2018YFB180012, the National Nature Science Foundation of China under 61971267, 61972223, 61861136003, and 61621091, Beijing Natural Science Foundation under L182038, Beijing National Research Center for Information Science and Technology under 20031887521, and research fund of Tsinghua University-Tencent Joint Laboratory for Internet Innovation Technology.

References

[1] D. Deng, C. Shahabi, U. Demiryurek, L. Zhu, R. Yu, and Y. Liu, "Latent space model for road networks to predict time-varying traffic," in *KDD*, 2016.

[2] Y. Gong, Z. Li, J. Zhang, W. Liu, Y. Zheng, and C. Kirsch, "Network-wide crowd flow prediction of sydney trains via customized online non-negative matrix factorization," in *CIKM*, 2018.

[3] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, "Multi-task representation learning for travel time estimation," in *KDD*, 2018.

[4] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," *AAAI*, 2019.

[5] F. Monti, M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," in *NIPS*, 2017.

[6] P. Borgnat, C. Robardet, P. Abry, P. Flandrin, J.-B. Rouquier, and N. Tremblay, "A dynamical network view of lyons vélov shared bicycle system," in *Dynamics On and Of Complex Networks, Volume 2*. Springer, 2013, pp. 267–284.

[7] H. Shi, C. Zhang, Q. Yao, Y. Li, F. Sun, and D. Jin, "State-sharing sparse hidden markov models for personalized sequences," in *KDD*, 2019.

[8] B. Pan, U. Demiryurek, and C. Shahabi, "Utilizing real-world transportation data for accurate traffic prediction," in *ICDM*, 2012.

[9] G. Ristanoski, W. Liu, and J. Bailey, "Time series forecasting using distribution enhanced linear regression," in *PAKDD*, 2013.

[10] Y. Duan, Y. Lv, and F.-Y. Wang, "Travel time prediction with lstm neural network," in *ITSC*, 2016.

[11] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *ICLR*, 2018.

[12] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NIPS*, 2016.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2014.