https://doi.org/10.1016/j.trc.2021.103063

Joint predictions of multi-modal ride-hailing demands: a deep multi-task multi-graph learning-based approach

Jintao Ke[a], Siyuan Feng[b], Zheng Zhu[1b], Hai Yang[b], Jieping Ye[c]

[a]*Department of Logistic and Maritime Studies, Hong Kong Polytechnic University, Kowloon, Hong Kong, China*
[b]*Department of Civil and Environmental Engineering, the Hong Kong University of Science and Technology, Kowloon, Hong Kong, China*
[c]*Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, United States*

Abstract

Ride-hailing platforms generally provide various service options to customers, such as solo ride services, shared ride services, etc. It is generally expected that demands for different service modes are correlated, and the prediction of demand for one service mode can benefit from historical observations of demands for other service modes. Moreover, an accurate joint prediction of demands for multiple service modes can help the platforms better allocate and dispatch vehicle resources. Although there is a large stream of literature on ride-hailing demand predictions for one specific service mode, few efforts have been paid towards joint predictions of ride-hailing demands for multiple service modes. To address this issue, we propose a deep multi-task multi-graph learning approach, which combines two components: (1) multiple multi-graph convolutional (MGC) networks for predicting demands for different service modes, and (2) multi-task learning modules that enable knowledge sharing across multiple MGC networks. More specifically, two multi-task learning structures are established. The first one is the regularized cross-task learning, which builds cross-task connections among the inputs and outputs of multiple MGC networks. The second one is the multi-linear relationship learning, which imposes a prior tensor normal distribution on the weights of various MGC networks. Although there are no concrete bridges between different MGC networks, the weights of these networks are constrained by each other and subject to a common prior distribution. Evaluated with the for-hire-vehicle datasets in Manhattan, we show that our proposed approach outperforms the benchmark algorithms in prediction accuracy for different ride-hailing modes.

Keywords: ride-hailing, demand prediction, deep multi-task learning, multi-graph convolutional network

1. Introduction

Ride-hailing services, offering customers door-to-door ride services at any time and anywhere, have experienced explosive growth in recent years. One advantage of ride-hailing companies over traditional street-hailing taxi companies is that ride-sourcing companies can track and record the real-time trip information from both passenger side and driver side. Based on this information, the platform can discover the representative demand-supply patterns and predict passenger demand over time and over space (number of ride requests originating from one specific zone during one time interval). An accurate short-term prediction of passenger demand serves as a foundation of many operating strategies that aim to improve system efficiencies, such as surge pricing, vehicle dispatching, and vacant vehicle re-allocation, etc.

Most of the existing studies focus on predicting region-level ride-hailing passenger demand for one service mode (Yao et al., 2018, 2019; Ke et al., 2017; Geng et al., 2019b,a). They partition the examined city into various regular regions (squares or hexagons) or irregular regions (on the basis of administrative or geographical properties) and predict the near-future passenger demand in each region. However, in actual operations, ride-hailing companies commonly provide diversified ride services to customers with different interests. For example, solo ride services (such as UberX, Lyft, Didi Express), which dispatch one vehicle to serve one passenger at each time, are preferable by customers who are more inclined to time or feel uncomfortable about sharing rides with others. By contrast, shared ride services (such as

[1]Corresponding author. E-mail address: zhuzheng@ust.hk.

UberPool, Lyft Shared, Didi ExpressPool), which allow one driver to pick-up and drop-off two or more passengers in each ride with a discounted trip fare, are provided to customers who are more inclined to money. Some platforms even provide luxury ride services, such as Uber Black, to customers who are willing to pay for a better car environment. Moreover, many passengers do not stick to one service mode; instead, they may switch among different service modes in different circumstances (Lavieri and Bhat, 2019). For example, during peak hours, due to supply limitations, the platforms may implement surge pricing and raise the trip fare, such that passengers are more prone to use shared rides with a relatively low trip fare. This indicates that demands for different service modes interact with each other, thereby the historical observations of demands for one service mode can provide valuable information to the prediction of demands for other service modes.

Meanwhile, the platforms also have a strong desire for an accurate joint prediction of demands for multiple service modes, which help them better allocate and dispatch vehicle resources. For example, when the platform predicts that the demand for solo ride services will be substantially greater than the supply of regular vehicles in one region and there are sufficient idle luxury vehicles nearby, it can mitigate passenger queuing by dispatching luxury vehicles to serve solo ride passengers with free service upgrades. However, although the spatial-temporal prediction for ride-hailing demand has been examined for many years, most of the previous studies focused on prediction for one specific service mode. It remains unsolved and challenging in how to provide an accurate joint prediction of multi-modal ride-hailing demands, by a unified approach that can simultaneously model the spatial-temporal dependencies and knowledge sharing across prediction tasks.

To tackle this challenge, this study proposes a novel multi-task multi-graph learning approach. The approach views the prediction of each ride-hailing mode demand as one task. For each task, we propose a multi-graph convolutional (MGC) network to capture the non-Euclidean spatial-temporal dependencies among different regions based on both geographical and semantic aspects. Multiple graphs are developed, including a distance graph that models the pair-wise distance between each two regions, a neighborhood graph that indicates whether two regions are adjacent to each other, a functionality graph that characterizes the functional similarity between each two regions, and a mobility pattern graph that describes the correlation of the historical demand trends between each two regions.

On the basis of various MGC networks, we design two multi-task learning structures to share knowledge across different spatial-temporal prediction tasks. The first one is the regularized cross-task (RCT) learning, which builds concrete crossed connections between the inputs and outputs of different tasks, such that prediction of one service mode demand can take advantage of information from other service modes. In the objective function, to avoid over-fitting issues due to model complexity, we penalize the inter-task weights and intra-task weights with different intensities. The second structure is the multi-linear relationship (MLR) learning. Instead of using inter-task weights to concretely link different tasks, MLR assumes that the intra-weights of various MGC networks are subject to a common tensor normal prior distribution. Therefore, weights in different networks are restrained by each other, and different tasks learn to share knowledge. Based on multi-modal demand prediction experiments with actual ride-hailing data in Manhattan, New York, the proposed framework outperforms the benchmark algorithms. In summary, this paper makes the following contributions:

- We propose a novel multi-task multi-graph learning approach to enable the joint prediction of multi-modal ride-hailing demands as well as other spatial-temporal joint prediction tasks.
- Two multi-task learning methods, namely RCT learning and MLR learning, are proposed to share knowledge across the MGC networks for different prediction tasks.
- We conduct extensive experiments on the actual ride-hailing dataset in Manhattan which contains both solo and shared ride services. We show that the proposed approach outperforms the state-of-art algorithms, and the use of multi-task learning structures can improve predictive accuracy in different spatial-temporal prediction tasks.

2. Literature review

The forecasting of ride-hailing demands belongs to the huge family of spatial-temporal predictions. In this section, we provide a thorough review of conventional and advanced approaches for spatial-temporal prediction of travel demand as well as other traffic states (such as flow, speed, and density). Of particular focus is the emerging multi-task learning-based approaches that enable us to predict multi-modal ride-hailing demands or other traffic-related measurements simultaneously.

2.1. Conventional spatial-temporal approaches

The prediction of short-term transportation measurements was brought to the academic field in 1979 when the autoregressive integrated moving average (ARIMA) model was introduced to predict traffic flows (Ahmed and Cook, 1979). The time series ARIMA approach has been refined over time (Levin and Tsao, 1980; Hamed et al., 1995; Billings and Yang, 2006). Other statistical models and machine learning models were also proposed to solve prediction problems of traffic flow, traffic incidents, and travel demand. Conventional prediction approaches include regressions (Kamarianakis et al., 2010; Battifarano and Qian, 2019), Kalman filtering models (Okutani and Stephanedes, 1984; Lu and Zhou, 2014), Bayesian network (BN) models (Zhu et al., 2016), Neural network models (Park and Rilett, 1998; Zheng et al., 2006), K-nearest neighbor algorithm (Tak et al., 2014), tensor factorization (Zhu et al., 2021) and so on.

The majority of these approaches treat the predicted transportation states as univariate time series, ignoring the nature of spatial correlations in transportation systems. Some researchers have considered spatial-temporal covariates into traditional approaches for traffic states and travel demand predictions. Yin et al. (2002) considered upstream time series traffic flows to predict downstream traffic states via a fuzzy-neural model. Sun et al. (2006) adopted a Gaussian BN model to predict near-future traffic flow with both local and upstream volumes. Zhu et al. (2019) incorporated the joint probability distributions of traffic flows at nearby sensor stations into traffic speed prediction. Spatial-temporal covariates were also utilized via conventional approaches for the predictions of travel time (Wu et al., 2004), rail demand (Jiang et al., 2014), metro demand (Ni et al., 2016), etc. Although conventional approaches have alleviated the difficulties in forecasting the stochasticity of transportation states, a common limitation is that only the nearby spatial information was included in these models. With traditional model structures and estimation algorithms, it can be difficult to incorporate useful distant information into predictions.

2.2. Deep learning spatial-temporal approaches

In recent years, deep learning-based approaches have been widely used in transportation state predictions. Designed for research tasks such as image recognition, convolutional neural networks (CNNs) are capable of capturing high-order spatial-temporal correlations in transportation prediction problems. Spatial-temporal transportation states are naturally regarded as a series of images by dividing the study area into small regions or zones. And following this approach, researchers have utilized CNNs in various prediction tasks, including speed evaluation (Ma et al., 2015), bike usage prediction (Zhang et al., 2016), ride-hailing demand-supply prediction (Ke et al., 2018) and so on. Recurrent neural networks (RNNs) and their extensions such as long short-term memory (LSTM) are well fit for processing time series data streams. Xu et al. (2017) applied LSTM to predict taxi demand in New York City. Some researchers integrated RNNs with CNNs to make full use of spatial-temporal information to forecast short-term ride-hailing demand (Ke et al., 2017), traffic flow (Wu and Tan, 2016; Yu et al., 2017) and bike flow (Zhang et al., 2018).

Based on but not limited to the mechanism of CNNs and RNNs, there have been extensions on the integrated deep learning algorithms. Liu et al. (2019) developed a contextualized spatial-temporal network, which captures a local spatial context, a temporal evolution context, and a global correlation context, to predict taxi demand. Geng et al. (2019a) proposed a spatial-temporal MCG (ST-MCG) model that utilizes non-Euclidean correlations for ride-hailing demand prediction. Based on an encoding-decoding structure between CNNs and ConvLSTMs, Zhou et al. (2018) developed an attention-based deep neural network to forecast multi-step passenger demand for bikes and taxis.

2.3. Multi-task learning-based approaches

The aforementioned conventional and advanced approaches greatly enhance the capability of urban-wise mobility prediction and evaluation. The superiority in prediction accuracy with a specific transportation state (e.g. traffic flow and travel demand) forecasting task has been demonstrated in previous studies. Since transportation states can be correlated with each other, researchers become interested in the simultaneous prediction of multiple states. For instance, joint-prediction of morning and evening commute demands may be more accurate than single demand predictions due to the positive correlation between the two types of commute demands.

In machine learning approaches, multi-task learning is a good solution to joint prediction problems. Multi-task learning is a paradigm that aims to leverage useful information contained in multiple learning tasks for improving the performance of various tasks (Zhang and Yang, 2017). A deep multi-task learning model attempts to learn the correlated representation in the feature layers and independent classifiers in the classifier layer without affecting the relationships of the tasks (Long et al., 2017). Nowadays, substantial research efforts are dedicated to the application of multi-task deep learning algorithms for the simultaneous prediction of correlated transportation states. Kuang et al. (2019) embedded the common features of taxi pickup demand and taxi dropoff demand via an attention-based LSTM model, and jointly predicted the two taxi demands via a 3D residual deep neural network. Geng et al. (2019b) proposed a modality interaction mechanism to learn the interactions among different region-wise graph representations in MGCs. Zhang et al. (2019) proposed a multi-task temporal CNN approach for zone-level travel demand prediction.

However, little efforts have been directed towards the joint prediction of demands for multiple service modes in ride-hailing systems. Concerning the correlations among different ride-hailing service modes, it is meaningful to explore suitable ways to share knowledge across the prediction tasks for various demands.

3. Preliminaries

In this section, we first give explicit definitions to several key concepts and then formulate the multi-modal ride-hailing demands prediction problem.

3.1. Region partition

It is a common way in the literature to partition the examined area into various regular rectangles. This allows easy implementations of stylized spatial-temporal prediction models, such as CNNs, RNNs, and combinations of CNNs and RNNs, etc. There are also some studies (e.g. Ke et al., 2018) dividing the examined city into various regular hexagonal grids since hexagons have an unambiguous neighborhood definition, a smaller edge-to-area ratio (smoother boundaries) and nice isotropic properties. However, some regulators and planners divide their cities into various irregular grids, according to their administrative and geographical properties. They may want to dispatch vehicles or make other decisions based on these irregular zones. In addition, they may only offer grid-level aggregate trip information. For example, the dataset used in this paper — the for-hire-vehicle dataset in Manhattan, New York City — only provides information on the origin and destination zone of each trip, while a total of 63 zones in Manhattan are partitioned based on zip codes. It is worth noting that many real-time operations, such as vehicle dispatching, rely on accurate information based on fine-grained zones. Fortunately, the administrative zones in Manhattan are fine-grained enough for these real-time operations. The average area of the administrative zones in Manhattan is $0.938km^2$, while the average area of zones used for vehicle dispatching is generally larger than $1km^2$. For example, Mao et al. (2020) propose a reinforcement learning model to redistribute vehicles from zones with redundant supply to zones with insufficient supply. In their experiment, they separate Manhattan into 8 zones, which certainly shows that their zones are much larger than our zones. Another example is Lin et al. (2018), which uses a multi-agent reinforcement learning to perform vehicle dispatching based on regular hexagon zones with a length of side equal to 0.7km (implying that the area is $1.273km^2$). In terms of the temporal dimension, each day is uniformly divided into intervals with equal length time slices (e.g. one hour).

On the basis of the administrative region partitions, we build a weighted graph with nodes referring to the zones and edges characterizing the inter-zone relationships; thereby, zones are fully connected with each other in this graph (i.e. any two nodes have a connection via a link). Let $G(V, E, \boldsymbol{A})$ denote the weighted graph, where V is the set of zones, E is the set of edges, and $\boldsymbol{A} \in \mathbb{R}^{|V|\times|V|}$ is the adjacent matrix with each element indicating the relationship between two zones.

3.2. Research problem

In this paper, we target at predicting multi-modal region-level ride-hailing passenger demands in a short time interval. Suppose the platform provides a total of M ride-hailing service modes (such as expresses, luxury, shared ride service, etc.). Let $x_{i,m}^t$ denote the number of passenger requests (passenger demand) for service mode m in zone i during time interval t, and $\boldsymbol{X}_m^t$ denote passenger demands for service mode m in all zones at time interval t. As examined in many previous studies (e.g. Ke et al., 2017; Geng et al., 2019a; Yao et al., 2019) the problem of region-level ride-hailing demand prediction for one service mode m can be formulated as a single-task problem as follows,

Definition 1. (ride-hailing demand prediction) Given the historical observations of ride-hailing demand for service mode m before the current time interval t, that is $[\boldsymbol{X}_m^{t-T}, ..., \boldsymbol{X}_m^t]$, the problem is to predict the spatial-temporal ride-hailing demand for service mode m in the next time interval, that is, $\boldsymbol{X}_m^{t+1}$. T is the number of historical time intervals used for the prediction.

As aforementioned, it is naturally expected that demand prediction for one mode can benefit from the historical observations of demands for other modes. With this knowledge in mind, we formulate a multi-task learning problem that simultaneously predicts ride-hailing demands for all service modes by taking advantage of the historical demands for all service modes. The problem is formally defined as,

Definition 2. (multi-modal ride-hailing demands prediction) Given the historical observations of ride-hailing demands for service modes $[\boldsymbol{X}_m^{t-T}, ..., \boldsymbol{X}_m^t], \forall m \in \{1, ..., M\}$, the problem is to forecast the spatial-temporal ride-hailing demand for multiple service modes $\boldsymbol{X}_m^{t+1}, \forall m \in \{1, ..., M\}$.

As pointed out by Zhang and Yang (2017), one important issue in multi-task learning is how to share knowledge among various tasks. In what follows, we will present a multi-task multi-graph learning approach that spells out the concrete ways to share knowledge among different service modes for a better multi-modal demand prediction.

4. A deep multi-task multi-graph learning approach

In our proposed approach, we first capture both geographical and semantical non-Euclidean relationships among zones in multiple graphs. It is worth mentioning that the graphs for different service modes are not identical, since some graphs characterize the mobility patterns (trends of historical demand), which are different across service modes. For each service mode, we then implement an MGC network to predict its region-level (i.e. zone-level) demand on the basis of its corresponding graphs. Finally, we propose two multi-task learning structures, the RCT learning and MLR learning, that specify the ways to share knowledge across different tasks (namely, predictions for different service modes).

4.1. Spatial dependence and multi-graphs

In an MGC network, geographical and semantic relationships among zones are represented by the graph structure and its associated adjacent matrices. Now we construct three common graphs that are shared by all service modes (the neighborhood graph $G_N(V, E, \boldsymbol{A}_N)$, distance graph $G_D(V, E, \boldsymbol{A}_D)$, and functionality graph $G_F(V, E, \boldsymbol{A}_F)$), and one specific graph that is diverse across different service modes, i.e. the mobility pattern graph $G_P^m(V, E, \boldsymbol{A}_P^m)$. Formally, $\boldsymbol{A}_N$ and $\boldsymbol{A}_D$ are given by,

$$[\boldsymbol{A}_N]_{i,j} = \begin{cases} 1, \text{ if zone i and j are adjacent} \\ 0, \text{ otherwise} \end{cases} \tag{1}$$

$$[\boldsymbol{A}_D]_{i,j} = \frac{1}{Dist(lng_i, lat_i, lng_j, lat_j)} \tag{2}$$

where lng_i, lat_i are the longitude and latitude of the central point of zone i, $Dist(\cdot)$ calculates the straight-line distance between point (lng_i, lat_i) and (lng_j, lat_j), $[\boldsymbol{A}_N]_{i,j}$ refers to the element of adjacent matrix A_N in the ith row and jth column. Clearly, the shorter the straight-line distance between the centers of two zones, the larger the weight associated with these two zones in the distance graph (the stronger the relationship). These two graphs can well capture the pair-wise geographical relationships between zones.

In addition to having geographical relationships, different zones may be correlated with each other in a semantic manner. Usually, zones in a city have different functionalities or land-use properties:

some are business zones, while others are residential zones. The ride-hailing demands in two zones with similar functionalities can be strongly correlated, even though they are far away from each other geographically. With this knowledge in mind, we formulate the functionality graph by,

$$[\boldsymbol{A}_F]_{i,j} = \frac{1}{\sqrt{(\boldsymbol{s}_i - \boldsymbol{s}_j)(\boldsymbol{s}_i - \boldsymbol{s}_j)^T}} \tag{3}$$

where $\boldsymbol{s}_i$, $\boldsymbol{s}_j$ are the vector of functionalities of zone i and j. The vector of each zone includes the number of households with zero private cars, the density of houses, the density of population, the density of employments, lengths of road network per square kilometers, and average distances to metro stations, etc. These features can reflect the functionalities of zones. For example, zones with a larger density of houses could be residential areas; while zones with a larger density of employments could be commercial areas. All of these features are retrieved from the Smart Location Database (https://www.epa.gov/smartgrowth/smart-location-mapping) provided by the United States Environmental Protection Agency. This database includes more than 90 geographical attributes available for every census block group in the US. The attributes include housing density, destination accessibility, neighborhood design, diversity of land use, transit service, employment, and demographics, etc. It can be clearly seen from Eq. 3 that, the similar/closer the two vector of functionalities in zone i and j, the larger the value of $[\boldsymbol{A}_F]_{i,j}$, which implies a stronger relationship between zone i and j in terms of functionalities. Then the matrix of $[\boldsymbol{A}_F]$ redefine the pair-wise distances between each two zones in a semantic manner, and thus can induce the graph neural networks to capture the local spatial correlations between zones with similar functionalities.

It is also generally expected that zones with similar mobility patterns (represented by historical demand trends) may share common characteristics and provide useful predictive information to each other (Yao et al., 2018). Historical demand trends are different across service modes, and therefore we establish mode-specific mobility pattern graphs. For a specific service mode m, we have,

$$[\boldsymbol{A}_P^m]_{i,j} = \frac{\mathrm{Cov}(q_i^m, q_j^m)}{\sqrt{\mathrm{Var}(q_i^m)\mathrm{Var}(q_j^m)}} \tag{4}$$

where q_i^m, q_j^m are the long-term historical trends (vectors) of ride-hailing demand for service mode m in zone i and j, respectively, $\mathrm{Cov}(\cdot, \cdot)$ calculates the correlation of two time series vectors, $\mathrm{Var}(\cdot)$ calculates the variance of one time series vector.

4.2. Multi-graph convolutions

In the past few years, researchers have developed various types of graph neural networks. These networks can be roughly categorized into two groups: spectral graph convolutional networks that transform signals from graph domain to Fourier domain through a graph Laplacian, and spatial graph convolution networks that directly operate in the graph domain. In this paper, we mainly consider the spectral convolutions. To efficiently transform signals, Defferrard et al. (2016) employed a Chebyshev polynomial to approximate the graph Laplacian, and Kipf and Welling (2016) further simplified the graph Laplacian by re-normalizing a first-order Chebyshev polynomial. The latter method has a neat mathematical form and is widely used in many applications, such as node classifications in scholar networks and link prediction in social networks. In the spirit of this work and on the basis of the aforementioned multi-graphs, we formulate an MGC in the prediction for service mode m by,

$$\begin{aligned} &\mathcal{F}_W^m(\boldsymbol{X}; \boldsymbol{A}_N, \boldsymbol{A}_D, \boldsymbol{A}_F, \boldsymbol{A}_P^m) \\ &= \sigma\left(\sum_{r\in\{N,D,F,P\}} \widehat{\boldsymbol{A}_r^m}\boldsymbol{X}\boldsymbol{W}_{r,m} + b_m\right) \end{aligned} \tag{5}$$

where $\boldsymbol{W}_{r,m} \in \mathbb{R}^{f_i \times f_o}, \forall r \in \{N, D, F, P\}$ are trainable weights, $\boldsymbol{X} \in \mathbb{R}^{|V|\times f_i}$ are input features, f_i and f_o are the input and output feature dimensions, $\sigma(\cdot)$ is an activation function, b_m is the intercept. Matrix $\widehat{\boldsymbol{A}_r^m}$ is determined before training and given by,

$$\widehat{\boldsymbol{A}_r^m} = (\boldsymbol{D}_r^m)^{-1/2}\,\widetilde{\boldsymbol{A}_r^m}\,(\boldsymbol{D}_r^m)^{-1/2} \tag{6}$$

where $\widetilde{\boldsymbol{A}_r^m} = \boldsymbol{A}_r^m + \boldsymbol{I}$ is the sum of adjacent matrix and an identity matrix to ensure that each node takes advantage of the historical observations of itself. $\boldsymbol{D}_r^m$ is the degree matrix, where $[\boldsymbol{D}_r^m]_{ij} = \sum_j [\widetilde{\boldsymbol{A}_r^m}]_{ij}$. It can be shown that our MGC assigns different weights to multiple graphs, and uses the sum of the outputs of multiple graphs to generate the final output, in each service mode. Therefore, in one single graph convolution, we treat all trainable weights (for different graphs) as one weight matrix $\boldsymbol{W}_m = [..., \boldsymbol{W}_{r,m}, ...] \in \mathbb{R}^{\tilde{f}_i \times f_o}$, where $\tilde{f}_i = f_i * 4$.

4.3. Regularized cross-task learning

In this section, we propose a novel RCT learning structure that enables the predictions of different service modes to share knowledge with each other. To elaborate the key idea of RCT, we use Fig. 1 as a demo, in which two basic three-layer networks are established to predict the ride-hailing demand for two service modes (mode 1 in blue color may represent solo service and mode 2 in red color may denote shared service). Let $\boldsymbol{W}_{m\to n}^l$ denote the trainable weight matrix (containing trainable weights for all graphs as mentioned above) that is associated with a graph convolution operation from service mode m to service mode n in the lth layer. Without knowledge sharing (single-task learning), the network on the left directly maps the features of service mode 1 to its labels through two trainable weights $\boldsymbol{W}_{1\to 1}^1$ and $\boldsymbol{W}_{1\to 1}^2$; similarly, weights $\boldsymbol{W}_{2\to 2}^1$ and $\boldsymbol{W}_{2\to 2}^2$ are used to map the features of mode 2. This indicates that the networks for predicting different service modes are independent of each other.

In RCT learning, we design a cross-task structure among networks for different service modes. Mathematically, the output of the network for service mode m in layer l, denoted by $\boldsymbol{H}_m^{l+1}$ is given by,

$$\boldsymbol{H}_m^{l+1} = \sum_{k\in\{1,...,M\}} \mathcal{F}^k_{\boldsymbol{W}_{k\to m}^l}(\boldsymbol{H}_{\boldsymbol{k}}^{\boldsymbol{l}}; \boldsymbol{A}_N, \boldsymbol{A}_D, \boldsymbol{A}_F, \boldsymbol{A}_P^k) \tag{7}$$

where convolution operation $\mathcal{F}^k_{\boldsymbol{W}_{k\to m}^l}$ maps from $\boldsymbol{H}_k^l$, namely, the inputs of the network for service mode k in layer l, to $\boldsymbol{H}_m^{l+1}$, and is parameterized by $\boldsymbol{W}_{k\to m}^l$. We denote the weights that transform input to output within the same task as intra-task weights, and the weights that connect input and output of different tasks as inter-task weights. For example, in Fig. 1, $\boldsymbol{W}_{1\to 1}^1$ and $\boldsymbol{W}_{1\to 1}^2$ are intra-weights, while $\boldsymbol{W}_{1\to 2}^1$ and $\boldsymbol{W}_{2\to 1}^2$ are inter-weights. In this way, the prediction task for a service mode m can take advantage of the information not only from its own features, but also from features of other service modes. However, RCT learning may greatly increase the number of weights, particularly when there are many service modes. To address this problem, we penalize the weights in the objective function by introducing the following regularization term:

$$J_1^l = \alpha \sum_{i=1}^{M} \left\| \boldsymbol{W}_{i\to i}^l \right\|_2^2 + \sum_{i=1}^{M} \sum_{j=1, j\neq i}^{M} \left\| \boldsymbol{W}_{i\to j}^l \right\|_2^2 \tag{8}$$

where α is a pre-defined parameter that determines the trade-offs between the penalties of intra-weights and inter-weights. In general, α is set to be smaller than 1, indicating that a smaller penalty is imposed on intra-weights, as compared with inter-weights. The reason is that the prediction of future demand for a service mode benefits more from the historical observations of its own features, than features of other service modes.

Let $\mathcal{X}_m = \{\mathbf{X}_m^1, ..., \mathbf{X}_m^{N_m}\}$, $\mathcal{Y}_m = \{\mathbf{Y}_m^1, ..., \mathbf{Y}_m^{N_m}\}$ denote the training features and labels of task m (the predicted demand for service mode m), where N_m is the number of training samples of task m. In our problem, N_m is the total number of time steps to be predicted in the training dataset. Therefore, in a RCT learning framework, the parameters of the networks can be trained by solving the following problem:

$$\min_{\mathcal{W},\mathbf{b}} \sum_{m=1}^{M} \sum_{s=1}^{N_m} \left\| \hat{\mathbf{X}}_m^s - \mathbf{X}_m^s \right\|_2^2 + \beta_1 \sum_{l\in\mathcal{L}} J_1^l \tag{9}$$

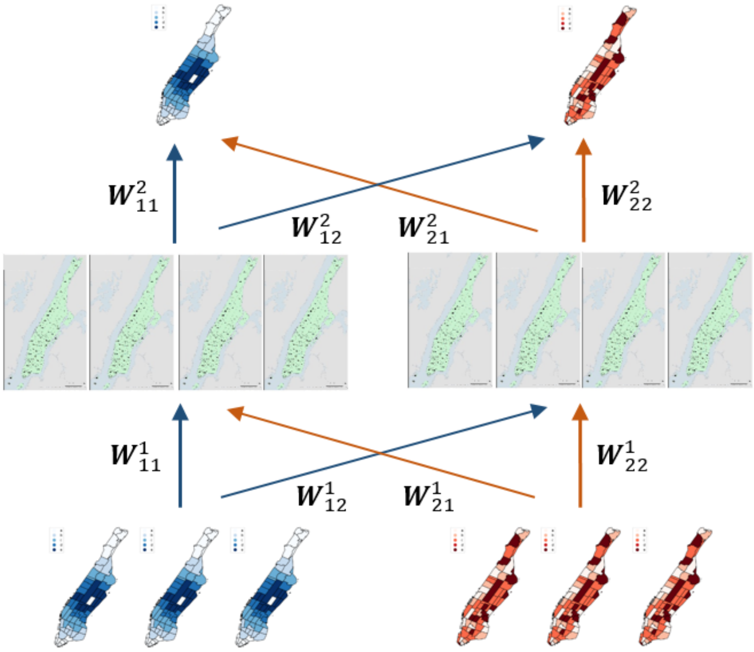


Figure 1: Regularized cross-task learning

where $\mathcal{L}$ is the set of layers, $\mathcal{W}, \mathbf{b}$ represent all weights and bias in parameters, $\hat{\mathbf{X}}_m^s$ is the predicted value for ground truth $\mathbf{X}_m^s$ by the neural networks, β_1 is a parameter balancing the trade-offs between bias and variance. The first term minimizes the squared loss between predicted demand and actual demand, while the second term is a regularized term given by Eq. 8.

4.4. Multi-linear relationship learning

In this section, we use an alternative weight to share knowledge across different tasks. As demonstrated in Fig. 2, instead of building cross connections between the inputs and outputs of networks for different service modes, we apply a MLR learning module (first proposed by Long et al. (2017)) that imposes a prior normal distribution on the intra-weights of multiple networks. This indicates that the intra-weights of different networks are constrained by each other and subject to a common prior probability distribution.

First, we place the weights of all networks in layer l in one tensor, denoted by $\mathcal{W}^l$, shown as follows:

$$\mathcal{W}^l = [\boldsymbol{W}_{1\to1}^l, \boldsymbol{W}_{2\to2}^l, ..., \boldsymbol{W}_{M\to M}^l] \in \mathbb{R}^{\tilde{f}_i \times f_o \times M} \tag{10}$$

where $\tilde{f}_i, f_o$ are the input and output dimensions of one weight matrix as defined in Section IV.B, M is the number of tasks (or service modes). Let $\mathcal{X} = \{\mathcal{X}_m\}_{m=1}^M$, $\mathcal{Y} = \{\mathcal{Y}_m\}_{m=1}^M$ denote the complete training data for all M tasks. Given $\mathcal{X}$ and $\mathcal{Y}$, the Maximum A Posterior (MAP) estimation of parameters $\mathcal{W} = [..., \mathcal{W}^l, ...]$ is

$$\begin{aligned} p(\mathcal{W}|\mathcal{X}, \mathcal{Y}) &\propto p(\mathcal{W}) \cdot p(\mathcal{Y}|\mathcal{X}, \mathcal{W}) \\ &= \prod_{l \in \mathcal{L}} p(\mathcal{W}^l) \cdot \prod_{m=1}^{M} \prod_{n=1}^{N_m} p(\mathbf{Y}_m^n | \mathbf{X}_m^n, \mathcal{W}^l) \end{aligned} \tag{11}$$

where the first term in the right-hand-side, $p(\mathcal{W}^l)$, is the prior, and the second term, $p(\mathbf{Y}_m^n|\mathbf{X}_m^n, \mathcal{W}^l)$, is a maximum likelihood estimation (MLE) given by the neural networks. We assume that the joint weight tensor $\boldsymbol{W}^l$ follows a tensor normal prior distribution as below,

$$\boldsymbol{W}^l \sim \mathcal{TN}_{\tilde{f}_i \times f_o \times M}(\overline{\boldsymbol{W}}^l, \Sigma^l) \tag{12}$$

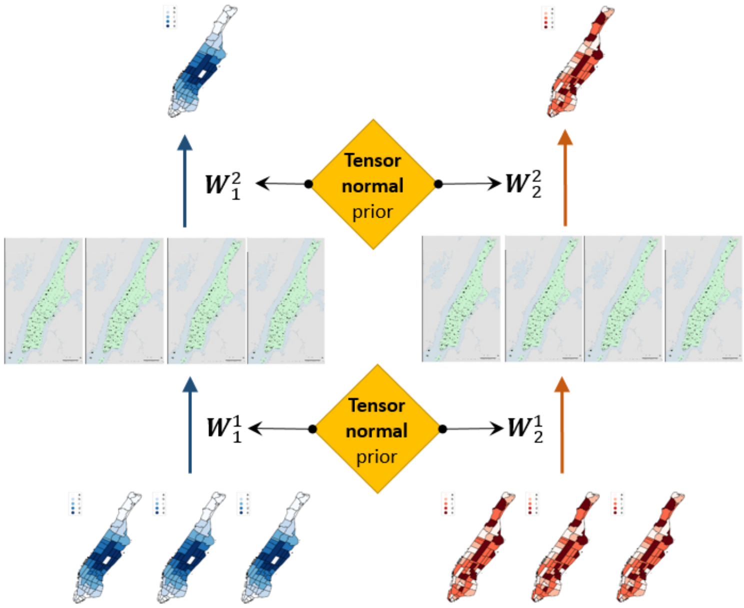


Figure 2: Multi-linear learning

where $\overline{\boldsymbol{W}}^l$ is the mean tensor, $\Sigma^l \in \mathbb{R}^{(\tilde{f}_i \cdot f_o \cdot M)\times(\tilde{f}_i \cdot f_o \cdot M)}$ is the covariance matrix. As pointed out by Long et al. (2017), this assumption in the prior term can well capture the multi-linear relationship across parameter tensors. The covariance matrix Σ^l may have an extreme large dimension, leading to computational difficulties. To address this issue, we decompose Σ^l into the Kronecker product of three small covariance matrices: $\Sigma^l = \Sigma_I^l \otimes \Sigma_O^l \otimes \Sigma_M^l$, where $\Sigma_I^l \in \mathbb{R}^{\tilde{f}_i \times \tilde{f}_i}$, $\Sigma_O^l \in \mathbb{R}^{f_o \times f_o}$, $\Sigma_M^l \in \mathbb{R}^{M \times M}$ are input covariance matrix, output covariance matrix, and service mode covariance matrix, respectively. The input covariance matrix Σ_I^l is computed by the covariance between the rows of the mode-1 matrix[2] of $\boldsymbol{W}^l$, i.e. $\boldsymbol{W}^l_{(1)} \in \mathbb{R}^{\tilde{f}_i \times (f_o \cdot M)}$. The other two covariance matrices Σ_O^l and Σ_M^l are computed in a similar way.

Substituting Eq. 12 into Eq. 11 and taking the negative logarithm give rise to the following regularized optimization problem:

$$\min_{\mathcal{W},\mathbf{b}} \sum_{m=1}^{M} \sum_{s=1}^{N_m} \left\| \hat{\mathbf{X}}_m^s - \mathbf{X}_m^s \right\|_2^2 + \frac{1}{2}\beta_2 \sum_{l \in \mathcal{L}} J_2^l \tag{13}$$

where β_2 is a parameter balancing the trade-offs between bias and variance, the regularized term J_2^l in layer l is given by,

$$\begin{aligned} J_2^l =& \mathrm{vec}(\boldsymbol{W}^l)^T (\Sigma_I^l \otimes \Sigma_O^l \otimes \Sigma_M^l)^{-1} \mathrm{vec}(\boldsymbol{W}^l) \\ &- \frac{D}{\tilde{f}_i} \ln(|\Sigma_I^l|) - \frac{D}{f_o} \ln(|\Sigma_O^l|) - \frac{D}{M} \ln(|\Sigma_M^l|) \end{aligned} \tag{14}$$

where $D = \tilde{f}_i \cdot f_o \cdot M$. The covariance matrices Σ_I^l, Σ_O^l, Σ_M^l are updated with the flip-flop algorithm (Ohlson et al., 2013), during training process. In addition, we can fix Σ_I^l and/or Σ_O^l (for example, assigned with identity matrices) and do not update their values during the training process to increase training stability. In this condition, the model only focuses on knowledge sharing across different tasks. Moreover, it can be found that the regularized terms in optimization problems 9 and 13 are layer separable. Therefore, we can design a multi-layer network that shares knowledge across tasks, with RCT learning in some layers and MLR learning in other layers. Mathematically, we can formulate a flexible network below,

[2]The jth row of mode-k matrix of the tensor $\boldsymbol{W}^l$, i.e. $\boldsymbol{W}^l_{(k)}$, contains all elements of $\boldsymbol{W}^l$ with the kth index equal to j.

$$\min_{\mathcal{W},\mathbf{b}} \sum_{m=1}^{M} \sum_{s=1}^{N_m} \left\| \hat{\mathbf{X}}_m^s - \mathbf{X}_m^s \right\|_2^2 + \beta_1 \sum_{l \in \mathcal{L}_c} J_1^l + \frac{1}{2}\beta_2 \sum_{l \in \mathcal{L}_m} J_2^l \tag{15}$$

where $\mathcal{L}_c$ and $\mathcal{L}_m$ are the set of layers using RCT and MLR learning, respectively.

5. Experimental results

5.1. Data and models

In September 2018, New York TLC released the new for-hire-vehicle data, which was reported by transportation network companies such as Uber and Lyft. The dataset includes detailed pick-up and drop-off time (on a basis of a second) of the passengers as well as the TLC zone based pick-up and drop-off locations. In the dataset, there is a field representing the service mode of the trip, i.e., a solo ride or a shared ride. Based on this dataset, we summarize zone based hourly demand for both solo rides and shared rides in Manhattan (63 TLC zones in total). Fig. 3 illustrates the highly stochastic trend of daily demand for the two service modes in the year 2018.

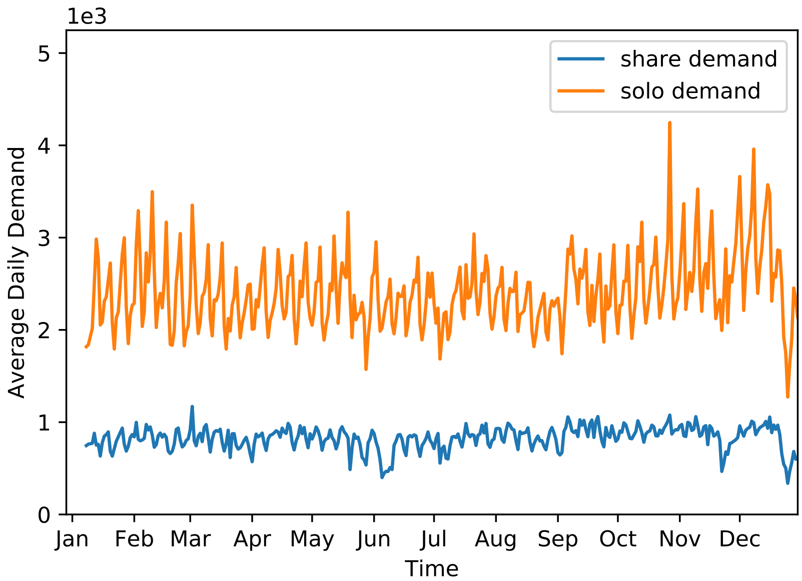
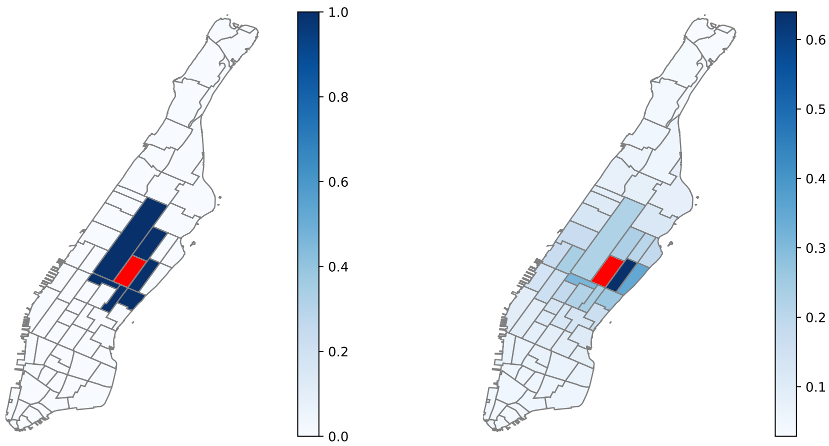


Figure 3: Time series of Manhattan ride-hailing demand

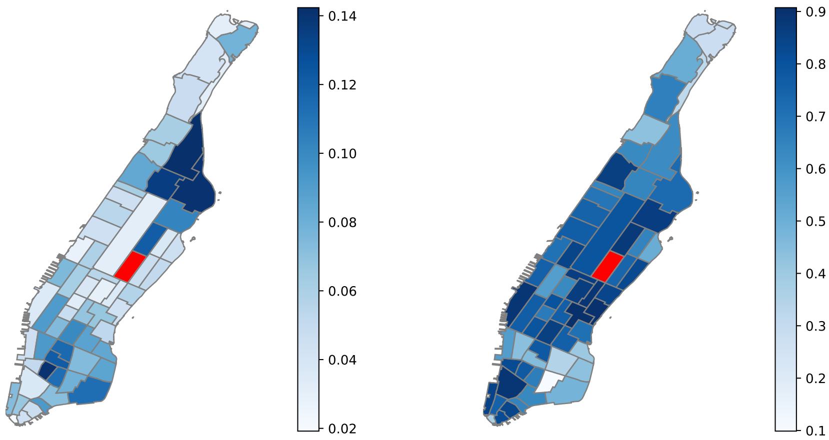
The spatial-temporal ride-hailing demand dataset is fused with land use attributes via another open source dataset – Smart Location Database. As aforementioned, this dataset is used to calculate pair-wise semantic relations between zones in terms of functional similarity.

With the aforementioned spatial dependence (i.e. graphs), Fig. 4 presents the multi-graph of zone 237 as an example. The target zone (id 237) is marked with red color. All the adjacent zones are highlighted in Fig. 4a; the darker the color of a zone, the stronger the relationship between this zone and the target zone. The distance graph is shown in Fig. 4b, in which zones closer to 237 have a higher value. The functionality graph calculated in Eq. 3 is illustrated in Fig. 4c, and the spatial correlation of shared service demand is shown in Fig. 4d. Neighbor and distance can only capture the spatial dependence of nearby zones; unlikely, some distant zones may have a strong correlation in terms of functionality or service demand pattern. These adjacent matrices can provide useful information for the spatial-temporal predictions in many different ways. For example, if only geographical information defined by neighbor graph and distance graph is used, the GCNs will only take advantage of the demand information in the surrounding zones as they implement predictions for the target zone. However, when the functionality graph is used as an adjacent matrix, the GCNs are able to utilize demand information in those zones with similar functionalities to forecast demand in the target zone.

In this real-world experiment, we use the demand data from 8 January 2018 to 4 November 2018 for models' training, 5 November 2018 to 2 December 2018 for models' validation, and 3 December 2018 to 31 December for models' testing. We compare different state-of-art machine learning approaches



(a) neighbor graph (b) distance graph



(c) functionality graph (d) shared graph

Figure 4: Graphs of zone 237

with the proposed deep multi-task learning approaches in terms of prediction accuracy. The models considered in this paper are described below:

- **HA**(historical average): HA directly predicts the future demand by the mean of historical demand of the same zone and the same interval in the past four weeks. HA is selected as the most straightforward and simple benchmark as a reference point.
- **LASSO** (Least Absolute Shrinkage and Selection Operator): LASSO (Tibshirani, 1996) is a generalized linear regression with an additional L1-norm regularization terms to avoid over-fitting. Since our problem is naturally a regression problem, we select this classical model as a baseline.
- **RF**(random forest): RF (Breiman, 2001) is a classical ensemble learning algorithm that constructs a multitude of decision trees at the training period and outputs the mean of the outputs of individual trees at the testing period. Due to its robustness and ability to avoid over-fitting issues, RF is widely used in many classification/regression tasks.
- **GBDT**(gradient boosting decision tree): GBDT (Friedman, 2001) generates the prediction by an ensemble of weak predictors, typically decision trees. GBDT is a classical gradient boosted machine

that has been widely used as benchmark algorithms for travel demand forecasting problems (Geng et al., 2019a).

- **XGB**(XGBoost): XGB (Chen and Guestrin, 2016) is a scalable, efficient, flexible and portable library for implementing machine learning algorithms under the Gradient Boosting framework. XGB is widely known as an efficient machine learning algorithm that can solve many data science problems in a fast and accurate way. In particular, it achieves outstanding performance in many machine learning competitions like Kaggle (www.kaggle.com).

- **MLP**(multi-layer perception): the MLP in our study simply uses a four-layer architecture, with one input layer, one output layer and two hidden layers. The Relu activation is used for the input layer and hidden layers, while Linear activation is used for the output layer. MLP is the most basic neural network and widely used as a benchmark algorithm in previous studies (Ke et al., 2018).

- **MGC**(multi-graph convolutional networks): MGC is first used by Geng et al. (2019a) for ride-sourcing travel demand forecasting, and demonstrates remarkable performance in experiments based on Didi's mobility data.

- **RCT-MGC**: a deep learning model that uses two symmetric four-layer MGC networks (with 128, 256, 128, and 1 units) for the two prediction tasks (solo and shared service demand). The four layers in the two networks are connected with RCT modules.

- **MLR-MGC**: a deep learning model that builds a similar structure as RCT-MGC, except that the four layers in the networks for the two tasks share knowledge with each other with MLR.

- **MIX-MGC**: a deep learning model that has a similar structure with RCT-MGC, except that the two lower layers share knowledge through RCT and the two upper layers share knowledge through MLR. The reason for using RCT in two lower layers and MLR is two upper layers is that features will become more and more generalized from bottom layers to upper layers. By creating concrete connections between the two tasks, RCT can better extract specific spatial features by completing graph connectivity, while MLR is more suitable for learning more generalized (abstract) features in upper layers (Geng et al., 2019b). This mixed structure may take advantage of the ability of RCT in capturing specific features and the ability of MLR in capturing generalized features simultaneously.

Table 1: Structure and hyper Parameters of MGC networks

Hyper parameters	MGC	RCT-MGC	MLR-MGC	MIX-MGC
Number of units in hidden layers	128, 256, 128	128, 256, 128	128, 256, 128	128, 256, 128
Optimizer	Adam	Adam	Adam	Adam
Learning rate	0.001	0.001	0.001	0.001
Activation Function	Relu	Relu	Relu	Relu
Number of Epochs	300	300	300	300
α	-	0.1	-	0.1
β_1	-	0.001	-	0.001
β_2	-	-	0.1	0.1

The parameters of the MGC networks are presented in Table 1. For a fair comparison, we use the same network structure for the two prediction tasks in all MGC networks, while learning rate, activation function, and the number of epochs are set to be the same for different MGC networks. In RCT-MGC, the hyperparameter α is 0.1 to impose a relatively small penalty on the intra-weights, and a relatively large penalty on the inter-weights. The two balancing factors in the objective function β_1 and β_2 are 0.001 and 0.1 respectively. The neural networks are implemented using PyTorch with a batch size of 16. The parameters of all the abovementioned benchmark algorithms are fine-tuned. Each model is fed with features including $\boldsymbol{X}_m^{t-1}$, $\boldsymbol{X}_m^{t}$ (the most recent two historical demands), $\boldsymbol{X}_m^{t+1-24}$ (historical demands during the same hour on yesterday), and $\boldsymbol{X}_m^{t+1-24\times 7}$ (historical demands during same hour on last week). All experiments are implemented on a server with 64G RAM and one NVIDIA 1080Ti GPU.

5.2. Results on the testing dataset

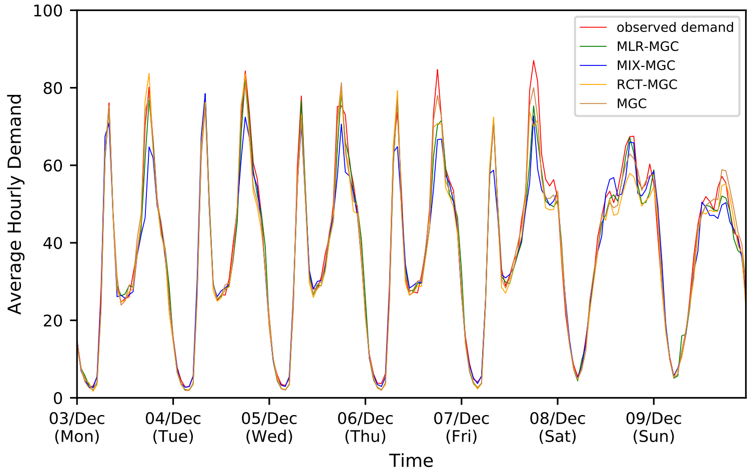
We examine the prediction error of the models by three measurements, Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). Since a zero observed hourly demand will drive MAPE to infinity, we only include the data records with positive demand for the calculation of MAPE. The performances of the models are depicted in TABLE 2. For both solo service demand and shared service demand, the four deep learning models significantly outperform the benchmarks of conventional machine learning models. For instance, compared with the MLP model, the MGC model can reduce RMSE/MAE/MAPE by 12.4%/14.1%/11.7% for solo service demand prediction, and reduce the measurements by 10.0%/10.0%/17.5% for shared service demand prediction. This indicates that the spatial correlations (i.e., both Euclidean and non-Euclidean dependencies) provide important information in spatial-temporal ride-hailing demand prediction; the correlations can be well characterized by the proposed adjacent matrices in the MGC modeling framework.

Moreover, based on the comparison between model MGC and models RCT-MGC, MLR-MGC and MIX-MGC, we note that a multi-task learning structure can further improve the prediction accuracy. The results indicate that demands of different ride-hailing service modes indeed have significant dependence, which can be captured via deep multi-task learning approaches. Additionally, we show that MLR-MGC and MIX-MGC perform slightly better than RCT-MGC in both solo service demand and shared service demand. The possible reason is that the features become highly generalized/abstract after a few layer transformations, while MLR is more capable of capturing correlations of generalized features between different tasks than RCT. Nevertheless, the architecture with two RCT layers and two MLR layers only bring about a very slight improvement in predictive performance, in comparison with the pure MLR structure. This implies that MLR's performance may overwhelm RCT's performance.

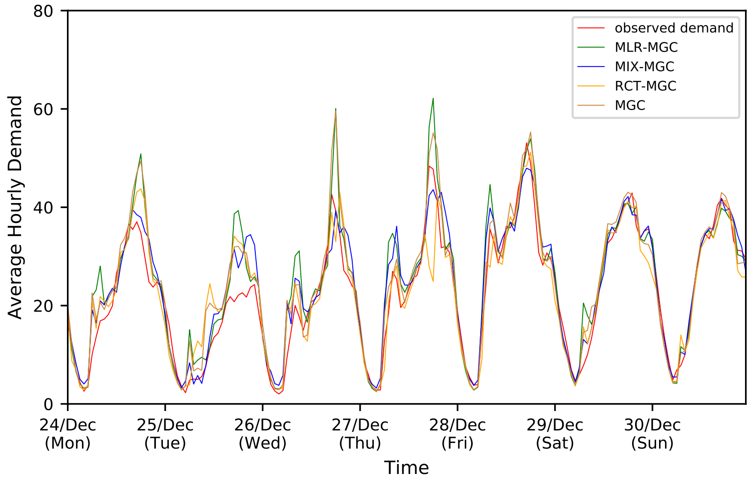
Table 2: Results of the testing dataset

demand of solo service rides			
Model	RMSE	MAE	MAPE
RCT-MGC	20.238	12.949	0.216
MLR-MGC	19.896	12.963	0.239
MIX-MGC	19.726	12.748	0.235
MGC	20.555	13.097	0.226
MLP	23.459	15.246	0.256
XGB	23.721	15.334	0.256
GBDT	23.806	15.365	0.256
RF	24.623	15.908	0.260
LASSO	26.906	17.365	0.308
HA	53.712	29.835	0.471
demand of shared service rides			
Model	RMSE	MAE	MAPE
RCT-MGC	9.316	6.059	0.322
MLR-MGC	8.937	5.994	0.343
MIX-MGC	8.727	5.919	0.343
MGC	9.536	6.346	0.350
MLP	10.595	7.050	0.424
XGB	10.621	6.999	0.401
GBDT	10.670	7.017	0.401
RF	11.187	7.405	0.420
LASSO	12.465	7.986	0.476
HA	19.227	10.931	0.600

Fig. 5 depicts hourly prediction results of shared service demand versa real-world observations. It can be seen that in both regular days (Fig. 5a) and holidays (Fig. 5b), the deep learning models can largely forecast the upcoming demand. However, the MGC networks tend to overestimate or underestimate the demand, in some special periods, such as the Christmas Holiday. Generally, compared to pure MGC, the MGC networks integrated with multi-task learning modules overestimate/underestimate the fluctuating demand to a smaller extent. We can also observe that the MGC networks perform better on a regular day than holiday, since demand uncertainty and fluctuation are larger on a holiday. This is normal since our models are partially fed with periodicity features, such as demand of the same time interval and the same zone in the last day, which may lead the predictions to follow the same patterns as the last day or last week. It is indeed a challenging problem to predict a sudden increase or decrease of demand, which merits more explorations in future research.



(a) Regular day



(b) Holiday day

Figure 5: Hourly prediction results

6. Conclusion

This paper studies the joint prediction of passenger demands for multiple service modes in ride-hailing systems. To enable effective knowledge sharing across different spatial-temporal prediction tasks, We propose a novel deep multi-task multi-graph learning approach, which first establishes separate MGC networks for different service modes, and then connects the networks with RCT and MLR learning techniques. While RCT learning builds up concrete bridges between different MGC networks, MLR learning imposes a soft connection among various MGC networks by assuming that their parameters follow a common prior probability distribution. Evaluated against a real-world ride-hailing dataset in Manhattan, we show that our proposed models significantly outperform the benchmark algorithms. Moreover, the use of multi-task learning techniques on the basis of MGC networks can further improve the prediction accuracy in spatial-temporal prediction tasks for multiple service modes. This study opens a few avenues that worth exploration, to name a few, (1) joint predictions of passenger demands for different transportation modes (such as bikes, private cars, and public transits); (2) joint predictions of passenger demand for ride-hailing services on multi-zone levels.

Acknowledgment

The work described in this paper was supported by a grant from Hong Kong Research Grants Council under project HKUST16208619 and a NSFC/RGC Joint Research grant N_HKUST627/18

(NSFC-RGC 71861167001). This work was also supported by the Hong Kong University of Science and Technology - DiDi Chuxing (HKUST-DiDi) Joint Laboratory.

References

Ahmed, M. S. and Cook, A. R. (1979). *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*. Number 722.

Battifarano, M. and Qian, Z. S. (2019). Predicting real-time surge pricing of ride-sourcing companies. *Transportation Research Part C: Emerging Technologies*, 107:444–462.

Billings, D. and Yang, J.-S. (2006). Application of the arima models to urban roadway travel time prediction-a case study. In *2006 IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2529–2534. IEEE.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Geng, X., Li, Y., Wang, L., Zhang, L., Yang, Q., Ye, J., and Liu, Y. (2019a). Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *2019 AAAI Conference on Artificial Intelligence (AAAI'19)*.

Geng, X., Wu, X., Zhang, L., Yang, Q., Liu, Y., and Ye, J. (2019b). Multi-modal graph interaction for multi-graph convolution network in urban spatiotemporal forecasting. *arXiv preprint arXiv:1905.11395*.

Hamed, M. M., Al-Masaeid, H. R., and Said, Z. M. B. (1995). Short-term prediction of traffic volume in urban arterials. *Journal of Transportation Engineering*, 121(3):249–254.

Jiang, X., Zhang, L., and Chen, X. M. (2014). Short-term forecasting of high-speed rail demand: A hybrid approach combining ensemble empirical mode decomposition and gray support vector machine with real-world applications in china. *Transportation Research Part C: Emerging Technologies*, 44:110–127.

Kamarianakis, Y., Gao, H. O., and Prastacos, P. (2010). Characterizing regimes in daily cycles of urban traffic using smooth-transition regressions. *Transportation Research Part C: Emerging Technologies*, 18(5):821–840.

Ke, J., Yang, H., Zheng, H., Chen, X., Jia, Y., Gong, P., and Ye, J. (2018). Hexagon-based convolutional neural network for supply-demand forecasting of ride-sourcing services. *IEEE Transactions on Intelligent Transportation Systems*.

Ke, J., Zheng, H., Yang, H., and Chen, X. M. (2017). Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies*, 85:591–608.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kuang, L., Yan, X., Tan, X., Li, S., and Yang, X. (2019). Predicting taxi demand based on 3d convolutional neural network and multi-task learning. *Remote Sensing*, 11(11):1265.

Lavieri, P. S. and Bhat, C. R. (2019). Investigating objective and subjective factors influencing the adoption, frequency, and characteristics of ride-hailing trips. *Transportation Research Part C: Emerging Technologies*, 105:100–125.

Levin, M. and Tsao, Y.-D. (1980). On forecasting freeway occupancies and volumes (abridgment). *Transportation Research Record*, (773).

Lin, K., Zhao, R., Xu, Z., and Zhou, J. (2018). Efficient large-scale fleet management via multi-agent deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1774–1783.

Liu, L., Qiu, Z., Li, G., Wang, Q., Ouyang, W., and Lin, L. (2019). Contextualized spatial-temporal network for taxi origin-destination demand prediction. *IEEE Transactions on Intelligent Transportation Systems*.

Long, M., Cao, Z., Wang, J., and Philip, S. Y. (2017). Learning multiple tasks with multilinear relationship networks. In *Advances in neural information processing systems*, pages 1594–1603.

Lu, C.-C. and Zhou, X. (2014). Short-term highway traffic state prediction using structural state space models. *Journal of Intelligent Transportation Systems*, 18(3):309–322.

Ma, X., Yu, H., Wang, Y., and Wang, Y. (2015). Large-scale transportation network congestion evolution prediction using deep learning theory. *PloS one*, 10(3):e0119044.

Mao, C., Liu, Y., and Shen, Z.-J. M. (2020). Dispatch of autonomous vehicles for taxi services: A deep reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, 115:102626.

Ni, M., He, Q., and Gao, J. (2016). Forecasting the subway passenger flow under event occurrences with social media. *IEEE Transactions on Intelligent Transportation Systems*, 18(6):1623–1632.

Ohlson, M., Ahmad, M. R., and Von Rosen, D. (2013). The multilinear normal distribution: Introduction and some basic properties. *Journal of Multivariate Analysis*, 113:37–47.

Okutani, I. and Stephanedes, Y. J. (1984). Dynamic prediction of traffic volume through kalman filtering theory. *Transportation Research Part B: Methodological*, 18(1):1–11.

Park, D. and Rilett, L. R. (1998). Forecasting multiple-period freeway link travel times using modular neural networks. *Transportation research record*, 1617(1):163–170.

Sun, S., Zhang, C., and Yu, G. (2006). A bayesian network approach to traffic flow forecasting. *IEEE Transactions on intelligent transportation systems*, 7(1):124–132.

Tak, S., Kim, S., Jang, K., and Yeo, H. (2014). Real-time travel time prediction using multi-level k-nearest neighbor algorithm and data fusion method. In *Computing in Civil and Building Engineering (2014)*, pages 1861–1868.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Wu, C.-H., Ho, J.-M., and Lee, D.-T. (2004). Travel-time prediction with support vector regression. *IEEE transactions on intelligent transportation systems*, 5(4):276–281.

Wu, Y. and Tan, H. (2016). Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. *arXiv preprint arXiv:1612.01022*.

Xu, J., Rahmatizadeh, R., Bölöni, L., and Turgut, D. (2017). Real-time prediction of taxi demand using recurrent neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 19(8):2572–2581.

Yao, H., Tang, X., Wei, H., Zheng, G., and Li, Z. (2019). Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *AAAI Conference on Artificial Intelligence*.

Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., and Li, Z. (2018). Deep multi-view spatial-temporal network for taxi demand prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yin, H., Wong, S., Xu, J., and Wong, C. (2002). Urban traffic flow prediction using a fuzzy-neural approach. *Transportation Research Part C: Emerging Technologies*, 10(2):85–98.

Yu, H., Wu, Z., Wang, S., Wang, Y., and Ma, X. (2017). Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. *Sensors*, 17(7):1501.

Zhang, J., Zheng, Y., Qi, D., Li, R., and Yi, X. (2016). Dnn-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 92. ACM.

Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X., and Li, T. (2018). Predicting citywide crowd flows using deep spatio-temporal residual networks. *Artificial Intelligence*, 259:147–166.

Zhang, K., Liu, Z., and Zheng, L. (2019). Short-term prediction of passenger demand in multi-zone level: Temporal convolutional neural network with multi-task learning. *IEEE Transactions on Intelligent Transportation Systems*.

Zhang, Y. and Yang, Q. (2017). A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*.

Zheng, W., Lee, D.-H., and Shi, Q. (2006). Short-term freeway traffic flow prediction: Bayesian combined neural network approach. *Journal of transportation engineering*, 132(2):114–121.

Zhou, X., Shen, Y., Zhu, Y., and Huang, L. (2018). Predicting multi-step citywide passenger demands using attention-based neural networks. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 736–744. ACM.

Zhu, Z., Peng, B., Xiong, C., and Zhang, L. (2016). Short-term traffic flow prediction with linear conditional gaussian bayesian network. *Journal of Advanced Transportation*, 50(6):1111–1123.

Zhu, Z., Sun, L., Chen, X., and Yang, H. (2021). Integrating probabilistic tensor factorization with bayesian supervised learning for dynamic ridesharing pattern analysis. *Transportation Research Part C: Emerging Technologies*, 124:102916.

Zhu, Z., Tang, L., Xiong, C., Chen, X., and Zhang, L. (2019). The conditional probability of travel speed and its application to short-term prediction. *Transportmetrica B: Transport Dynamics*, 7(1):684–706.