Multi-Modal Graph Interaction for Multi-Graph Convolution Network in Urban Spatiotemporal Forecasting

Xu Geng
Hong Kong University of Science and Technology
xgeng@connect.ust.hk

Xiyu Wu
Didi AI Labs, Didi Chuxing
wuxiyu@didichuxing.com

Lingyu Zhang
Didi AI Labs, Didi Chuxing
zhanglingyu@didichuxing.com

Qiang Yang
Hong Kong University of Science and Technology
qyang@cse.ust.hk

Yan Liu
University of Southern California
yanliu.cs@usc.edu

Jieping Ye
Didi AI Labs, Didi Chuxing
yejieping@didichuxing.com

ABSTRACT

Graph convolution network based approaches have been recently used to model region-wise relationships in region-level prediction problems in urban computing. Each relationship represents a kind of spatial dependency, like region-wise distance or functional similarity. To incorporate multiple relationships into spatial feature extraction, we define the problem as a multi-modal machine learning problem on multi-graph convolution networks. Leveraging the advantage of multi-modal machine learning, we propose to develop modality interaction mechanisms for this problem, in order to reduce generalization error by reinforcing the learning of multi-modal coordinated representations. In this work, we propose two interaction techniques for handling features in lower layers and higher layers respectively. In lower layers, we propose grouped GCN to combine the graph connectivity from different modalities for more complete spatial feature extraction. In higher layers, we adapt multi-linear relationship networks to GCN by exploring the dimension transformation and freezing part of the covariance structure. The adapted approach, called multi-linear relationship GCN, learns more generalized features to overcome the train-test divergence induced by time shifting. We evaluated our model on ride-hailing demand forecasting problem using two real-world datasets. The proposed technique outperforms state-of-the-art baselines in terms of prediction accuracy, training efficiency, interpretability and model robustness.

CCS CONCEPTS

- Computing methodologies → Multi-task learning, Neural networks.

KEYWORDS

- Multi-modal machine learning, Graph convolution networks, Multi-task learning, Transfer learning

ACM Reference Format:

Xu Geng, Xiyu Wu, Lingyu Zhang, Qiang Yang, Yan Liu, and Jieping Ye. 2019. Multi-Modal Graph Interaction for Multi-Graph Convolution Network in Urban Spatiotemporal Forecasting. In Proceedings of ACM Conference (Conference’17). ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/mmmm.mmmmmm

1 INTRODUCTION

The deployment of urban sensor networks is one of the most important progresses in urban digitization process. Recent advances in sensor technology enables the collection of a large variety of datasets. Multi-modality is one of the most significant features in knowledge discovery process in urban computing. Data from different sources are often correlated with each other. For region-level prediction problems, like crowd flow prediction [29, 30] or taxi demand prediction [6, 11, 21], it has become a common practice to incorporate a large variety of auxiliary datasets, like weather, POI, road network and events. In this paper, we define each auxiliary dataset as a modality and study multi-modal learning on multi graph convolution networks (MGCN) for spatiotemporal prediction problems in urban computing. This task is challenging due to complex spatial dependencies and temporal shifting generalization gap.

Designing spatial feature extraction method is challenging due to complex region-wise spatial dependencies. GCN-based models [15, 23] are first used for traffic prediction on road networks. Geng et al. [6] proposed Multi-GCN (MGCN) for generic spatiotemporal prediction tasks by stacking three GCNs. Each GCN encodes a unique modality (relationship) of auxiliary data (geo-distance, POI similarity and road network) as graph and extract spatial dependencies from such relationship. The spatial feature extraction by MGCN architecture is incomplete, due to the lack of cross-graph connectivities. Figure 1 shows an example for MGCN. Consider the vertex (region) pair A and D. According to graph topology, A and D are disconnected in all three graphs. MGCN is incapable of extracting features from D for A, or vice versa. However, we argue that the A→D relationship is important. The region pair A3→B3 and B2→D2 are closely related on road connectivity and POI similarity.
A and D are related region pairs for spatial feature extraction. To complete the physical meaning for spatial feature extraction by MGCN, the ideal graph connectivity is shown in 1(b). It is produced by merging all edges from separate graphs, so that any random walk path is a compound of any kind of relationships.

Improving model generality to overcome the temporal shifting generalization gap is another challenging task. Temporal pattern for time series data varies along with time. Formally,

$$P(X_t|X_{t-1}, X_{t-2}, \ldots) \neq P(X_{t'|t-1}, X_{t'-2}, \ldots), t \neq t'$$

The gap above defines the divergence between temporal pattern distributions in two different time windows $t$ and $t'$. Such a time shifting gap is often caused by time series fluctuations induced by periodicity, seasonality or miscellaneous factors like weather variation or events. We further discovered that this gap is usually accumulative. A longer temporal interval between two timestamps causes a larger divergence between two distributions. Due to this problem, machine learning models for time series prediction tasks expire frequently. Improving model generality makes the model more robust and avoid of fitting to local time series fluctuations.

We propose several graph interaction techniques to address to above problem, by enhancing the learning of multi-modal coordinated representations and reinforcing the model performance. Yosinski et al. [25] studied feature transferability in deep learning. Yao et al. [24] conducted experiments on two large-scale real-world datasets. The proposed approach achieves more than $10\%$ error reduction over state-of-the-art baseline methods for ride-hailing demand forecasting.

2 RELATED WORK

Region-level prediction in urban computing

Region-level prediction is a fundamental task in data-driven urban management. There are rich amount of topics, including citizen flow prediction [29, 30], traffic demand prediction [10, 11, 24], arrival time estimation [14] and meteorology forecasting [18, 19]. For these topics, the region-wise relationships are measured as geographical distance. The spatial structures for these prediction tasks are formulated as regular graphs, which are inherently euclidean structures. Convolution neural networks based models are used for effective prediction.

Non-euclidean structures exist in station-based prediction tasks, including bike-flow prediction [4], traffic volume prediction [15, 23, 26] and point-based taxi demand prediction [21]. The spatial structures for these problems are no longer regular. Graph convolution networks are usually leveraged for spatial feature extraction in these tasks. Non-euclidean structures also exist when incorporating auxiliary data to model region-wise relationships. Yao et al. [24] encoded region-wise relationship as a graph and use graph embedding as external features for convolution neural networks.
et al. [6] used MGCN to model region-wise relationships under multiple modalities.

**Multi-modality in urban computing**

The core issue for multi-modal machine learning is to build models that can process or relate information from multiple modalities [3]. Traditional multi-modal machine learning problems focus on human sensory modalities, including audio-visual speech recognition [28], multi-media analysis [2] and media description [9]. In urban computing, we usually need to harness knowledge from a diverse family of related datasets. Wei et al. [22] first categorized the diversity of urban computing datasets, such as POI and air quality as multi-modality and explored feature transferability among different modalities.

Multi-modal fusion is one of the most challenging problems in urban computing. Most existing works incorporate multi-modality auxiliary data as handcrafted features in a straightforward manner. Tong et al. [21] used multi-modality data as input features for linear regression model. Zhang et al. [29], Yao et al. [24] concatenated auxiliary data to high level abstractions for region-level spatiotemporal prediction networks.

GCN-based approaches encode multi-modality data as region-wise relationships and perform as a static structure in deep learning. The spatial feature extraction process on GCN is associated with these modalities. According to applications in traffic volume prediction [15] and taxi demand prediction [6], GCNs are effective in spatial feature extraction on spatial-variant modality data. However, all techniques above fail to build relationship among modalities, which is expected to improve the generality of the learning framework.

**Multi-task relationship learning**

Multi-task relationship learning is a basic approach for multi-task learning. Zhang and Yeung [32] first proposed a regularized multi-task relationship learning model MTRL by placing a task-variate normal prior on model parameter.

\[ W \sim MN(0, \Sigma_r, \Sigma_c) \]

where \( \Sigma_r \) and \( \Sigma_c \) are the row and column covariance. Long et al. [17] proposed Multilinear Relationship Network (MR Network) which learns multilinear relationship on different modes for the joint-task parameter tensor as:

\[ W = [W_1; W_2; \ldots; W_t] \]

\[ W \sim \mathcal{N}_{D_1 \times D_2 \times \ldots \times D_t}(\Sigma_f, \Sigma_c, \Sigma_t) \]

where \( W \) refers to the joint weight by concatenating all fully connected weights from all tasks. \( D_f, D_c \) and \( D_t \) denotes to the feature dimension, class dimension and task dimension in the joint weight. \( \Sigma_f, \Sigma_c \) and \( \Sigma_t \) represent covariance for each mode. Experiment results showed that imposing multi-task relationship regularizer on last few fully connected layers in CNN-like structures increased the feature generality and transferability in task specific layers.

However, MR Networks only learn multilinear relationships on fully connected layers. Other deep learning structures, like CNN or GCN have more complicated physical meanings.

---

**Table 1: Table of notations**

| Notation | Type | Meaning |
|----------|------|---------|
| \( R/V \) | set | Set of all regions (vertices) |
| \( M \) | set | Set of all modalities |
| \( K \) | scalar | Degree of chebyshev polynomial |
| \( d \) | \( \mathbb{R}^d \times d \) | Identity matrix with row/column size \( d \) |
| \( A_i \) | \( \mathbb{R}^{V \times V} \) | Adjacency matrix of \( i \)th modality |
| \( L_i \) | \( \mathbb{R}^{V \times V} \) | Symmetric normalized graph laplacian of \( i \)th modality |
| \( \Sigma_f \) | \( \mathbb{R}^{V \times 1} \) | (like ride-hailing demand) value at time \( t \) |
| \( X^t \) | \( \mathbb{R}^{V \times f} \) | \( f \)-dimensional feature of \( j \)th modality on \( l \)th layer |
| \( O^t \) | \( \mathbb{R}^{V \times 1} \) | Output layer as the \( l \)th layer |
| \( \sigma \) | function | Activation function |
| \( f_1, f_2 \) | scalar | Input feature dimension and output feature dimension |
| \( b_j \) | \( \mathbb{R}^{V \times f} \) | Bias for \( j \)th modality |
| \( W^l \) | \( \mathbb{R}^{K \times f_j \times f_i} \) | Weight of \( l \)th layer |
| \( w_{l, i,j} \) | \( \mathbb{R}^{K \times f_i \times f_j} \) | Weight for transforming \( X^l_j \) to \( X^l_{j+1} \) |
| \( w^l_{\alpha} \) | \( \mathbb{R}^{f_i \times f_j} \) | Weight corresponding to a specific chebyshev polynomial term |
| \( |I|, |O| \) | scalars | Input and Output dimension used to measure weight dimension |
| \( W^l_{i,a} \) | \( \mathbb{R}^{M \times K \times f_i \times f_j} \) | Weight of \( l \)th layer and \( a \)th chebyshev polynomial term of each mode in \( \Sigma \) |
| \( d \) | array of 4 | Dimension of \( d \)th mode |
| \( \Sigma_d \) | \( \mathbb{R}^{d_i \times d_j} \) | Covariance for the \( d \)th mode |
| \( \Sigma^l \) | \( \mathbb{R}^{\prod_{d \in [1]} d_i} \) | Kronecker decomposable covariance structure for tensor normal distribution |
| \( L_{high} \) | set | Higher layers assigned to multi-linear relationship GCN |
| \( L_{low} \) | set | Lower layers assigned to grouped GCN |

---

**3 METHODOLOGY**

Denote \( A = \{A_0, A_1, \ldots, A_M\} \) as adjacency matrices for different graphs. Each graph corresponds to one of the \( M \) modalities. In the ride-hailing demand prediction problem, each graph represents a kind of pair-wise spatial relationships for regions, including neighborhood (geo-distance) \( A_N \), POI similarity \( A_S \) and road connectivity \( A_C \) [6].

\[
A_{N,i,j} = \begin{cases}
1, & \text{if region } i \text{ and } j \text{ are adjacent} \\
0, & \text{otherwise}
\end{cases}
\]

\[
A_{S,i,j} = \text{sim}(P_{v_i}, P_{v_j})
\]

\[
A_{C,i,j} = \max(0, \text{conn}(v_i, v_j) - A_{N,i,j})
\]
$A_N$ defines adjacency relationship between regions. We construct $A_N$ by connecting a vertex to its 8 neighbors in a $3 \times 3$ grid. $A_S$ is the cosine similarity between POI vectors of two regions. Each entry in the POI vector represents the number of POIs in a specific category. $A_c$ indicates the connectivity between two regions. Two regions are connected as long as there is a highway or subway that directly connects them.

Define the one-step spatiotemporal prediction task for a certain modality ($A_i$) on a spatiotemporal observation $x$ as:

$$x_t = G(x_{t-1}, x_{t-2}, ..., x_{t-k}; A_i)$$  \(1\)

where $G$ represents any random-walk based graph convolution network. $x_t \in \mathbb{R}^{V}$ is the temporal slice of a spatiotemporal observation at time $t$.

When the graph convolution operation $G : (\mathbb{R}^{V|X'|}, \mathbb{R}^{V|X'|}) \rightarrow \mathbb{R}^{V|X'|}$ is defined as polynomial of graph laplacian $^1L$ with degree up to $K$:

$$G_W(X; A) = \sum_{\alpha=0}^{K} L^\alpha X W_\alpha$$  \(2\)

the above definition refers to the graph convolution operation of ChebNet [5]. In this work, we use this variation of graph convolution operations.

In multi-modality formulation of this problem, each modality refers to a representation learning process of the same spatiotemporal observation on different graphs. Following the convention in [3], we formalize joint a representation of multi-modality learning problem on multi-graph convolution network as:

$$x_t = \mathcal{F}_{A_i \in A}(G_W(x_{t-1}, x_{t-2}, ..., x_{t-k}; A_i))$$  \(3\)

where $\mathcal{F}_{A_i \in A}$ denotes the interaction function across multi-graphs. In previous work [6], it is defined as stacking function in anterior layers and sum function in the output layer. The major contribution of this work focuses on the design of this interaction function.

Figure 2 shows the proposed framework. According to analysis on feature generality [25] for deep neural networks, we proposed two techniques for building modality-wise interactions targeted for lower layers and higher layers respectively in stacked MGCNs. In lower layers, the hidden features are concrete. The feature extraction in lower layers are usually general. Considering these facts, we propose to build inter-modality connections to enable inter-graph spatial feature extraction. To distinguish feature extraction parameters, we penalize inter-graph weight and intra-graph weight differently by group regularization. In higher layers, the hidden features are highly abstract that they can no longer maintain their physical properties. Applying inter-modality connections is not applicable. High level features are usually task specific, which is harmful to model generality and transferability. In these layers, we propose to learn multilinear relationship on training parameters of joint modalities, in order to improve the model generality and avoid overfitting the model to local fluctuations.

Figure 2: Overview of the proposed graph interaction mechanism for stacked MGCNs. The multi-modality representation of input signals is generated by multi-graphs. In lower layers of deep neural networks, we use grouped GCN to enable inter-graph spatial feature extraction. In higher layers, we use multi-linear relationship GCN to learn modality-wise relationship by imposing tensor normal distribution on the joint representation of parameters. Finally we aggregate modalities to produce output.

**Grouped GCN**

Figure 3 shows one layer transformation of grouped GCN (GCGN). In lower layers, we use GCGN to build compound graph connectivity, which enables cross-graph spatial feature extraction.

Denote $L_i \in \mathbb{R}^{V|X'|}$ as the graph laplacian matrix of $i$-th modality. Denote $X_i^l \in \mathbb{R}^{V|X_i|}$ as the input signal of the $i$th modality of the $l$th layer. When $l = 1$, $X_i^1$ represents the raw input and $X_i^1 = X_j^1 \forall i,j$. Define the $l$th layer parameter $W_i^l$ as:

$$W_i^l = \left( \begin{array}{cccc} w_{i,1}^1 & w_{i,2}^1 & \cdots & w_{i,M}^1 \\ w_{i,1}^2 & w_{i,2}^2 & \cdots & w_{i,M}^2 \\ \vdots & \vdots & \ddots & \vdots \\ w_{i,M,1}^l & w_{i,M,2}^l & \cdots & w_{i,M,M}^l \end{array} \right),$$  \(4\)

1In this work, we use symmetric normalized laplacian $L = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$.

2$l, i \in \mathbb{Z}$
Denote $w_{i,j}^l \in \mathbb{R}^{d_i \times d_j \times K}$ as the weight matrix to transform the $i^{th}$ modality input to $j^{th}$ modality output via ChebNet transformation $G_{w_{i,j}^l}(X_i^l; A_l)$, where $d_i$ and $d_j$ are the feature dimension of $i^{th}$ and $(l + 1)^{th}$ layer. $K$ represents degree of Chebyshev polynomial, which is sliced during the computation of ChebNet. The $j^{th}$ modality output is computed as:

$$X_j^{l+1} = \sigma(\sum_{i=1}^{M} G_{w_{i,j}^l}(X_i^l; A_l) + b_j^l) \quad (5)$$

We denote all weights that transform input to output within same modality, i.e. $w_{i,j}^l$ for $i = j$, as intra-modality weights. Similarly, define the inter-modality weight as $w_{i,j}^l$ for $i \neq j$. It’s obvious that when all inter-modality weights are set to 0, the graph convolution operation defined above degrades to MGNC.

Adding cross-modality weights as stated above introduces a tremendous increment on the number of parameters with a factor of $O(M)$. This may boost the model complexity and cause overfitting. To address this issue, we use grouped sparsity [20, 27] to regularize the complexity of parameters. We designed flexible group regularization loss for layer $l$:

$$J_l^I = \alpha \sum_{i,j} ||w_{i,j}^l|| + \sum_{i,j} ||w_{i,j}^l|| \quad (6)$$

Different from traditional group regularization, we use a tunable parameter $\alpha$ to control the trade-off on penalties for intra-modality weights and inter-modality weights. To maintain the difference among modalities, we prefer a smaller $\alpha$ value, in order to introduce less penalty to intra-modality weight. The inter-modality feature extraction focuses on those highly strong relationships. This will help to maintain model generality from multi-modality throughout the proposed GGCN architecture.

The design strategy has several properties that maintain the advantage of GCN models. Firstly, the increment for computational complexity for GGCN is limited. The factor of time complexity increment is $O(M)$, which is polynomial of the number of modalities. In practice, the number of modalities are usually not large. Secondly, the extra computation above to compute intra-modality transformation and inter-modality transformation are naturally independent. It’s easy to design a parallel implementation. Finally, GGCN is a linear combination of different graph laplacians, which keep the numerical stability of the original MGNC model when using the normalized symmetric laplacian.

**Multi-linear relationship GCN**

In high level layers, latent features no longer maintain their properties as spatiotemporal observations. Instead of building cross-modality connections, we propose to learn multi-linear relationships (MR) on joint-modality weights by imposing tensor normal distribution as the prior distribution.

The dimensionality transformation of graph convolution operations in ChebNet is shown in Figure 4. There are totally five dimensions in the whole system, including regions/vertices ($R$, $|R| = |V|$), inputs (I), outputs (O), Chebyshev Polynomial (C, $|C| = K$) and modalities (M). For each single modality task, the representation of input signals on graph laplacian $L_i$ is in three dimensional space of region, input and chebyshev polynomial: $\{L_i^k \alpha = 0,1,...,K\} \in \mathbb{R}^{V \times |I| \times K}$. The model parameter for the $i^{th}$ modality is in three dimensional space of input, output and chebyshev polynomial: $W_i^l = \{w_{i,\alpha}^l = \{w_{i,\alpha} | \alpha = 0,1,...,K\} \in \mathbb{R}^{L_i^k \times |O| \times K}$.

The joint representation for multi-modality weight is defined as a four order tensor $W^l = \{W_l^1, W_l^2, ..., W_l^M\} \in \mathbb{R}^{L_i^k \times |O| \times K \times M}$.

Firstly, we impose tensor normal distribution as prior distribution for $W_l^l$:

$$W_l^l \sim \mathcal{T} \mathcal{N}_{L_i^k \times |O| \times K \times M}(M_l^l, \Sigma_l^l) \quad (7)$$

where $M_l^l$ is the mean tensor, $\Sigma_l^l = \Sigma_l^I \otimes \Sigma_l^O \otimes \Sigma_l^C \otimes \Sigma_l^M$ is the kronecker decomposable covariance structure. The density function is estimated as:

$$p(W_l^l) = 2\pi^{-\frac{1}{2}} \prod_{k=1}^{\frac{d}{2}} \left(\prod_{k=1}^{4} \left|\Sigma_k\right|^{-\frac{1}{2}} \prod_{k=1}^{\frac{d}{2}} \frac{d_k}{d_k!}\right) e^{-\frac{1}{2} (W_l^l - M_l^l)^T \Sigma_l^{-1} (W_l^l - M_l^l)} \quad (8)$$

where $d = |I| \times |O| \times K \times M$ represents dimensions for each mode, $\Sigma = [\Sigma_l^I, \Sigma_l^O, \Sigma_l^C, \Sigma_l^M]$, $|\cdot|$ represents determinant. According to Long et al. [17], for MAP estimation for model parameters, learning the posterior distribution of $W_l^l$ given training data $(X, Y)$ is equivalent to minimizing the negative logarithm for density of $\prod_j P(W_{l,j}^l)$, where $7$:

$$J_l^I = \frac{1}{2} (\text{vec}(W_l^l)^T \Sigma_l^{-1} \text{vec}(W_l^l)) \quad (9)$$

where $\text{vec}()$ is the flattening operation to transform a high-dimensional tensor to a 1-d vector. The flip-flop algorithm for updating covariance matrix of a certain mode $\Sigma_l$ is:

$$\Sigma_{d_i}^l = \frac{d_i}{\prod_{k=1}^{\frac{d}{2}} d_k} (W_l^l)_{i,j}(\otimes_{k \neq i} \Sigma_k)(W_l^l)_{i,j}^T + \epsilon I_{d_i} \quad (10)$$

}\[3\]

\[3\]Ignore terms irrelevant to $W_l^l$ because they have no gradient during back propagation.
where \( \epsilon_{ld} \) is a trade-off term for numerical stability. \((W^T h^{(i)})\) is the vectorization along the \( l^{th} \) mode. Such operation outputs matrix of shape \( \mathbb{R}^{(d_l) \times (I_l \times d_k)} \).

We further discovered that the covariance update rule above should not be applied to input (I) and output (O) modes. Instead, freezing covariance matrix of input (I) and output (O) modes to identity matrix \( I_d \) will improve model generality and transferability. Observing equation 2 of ChebNet on the \( l^{th} \) layer:

\[
G_W(X^l; A) = \sum_{\alpha=0}^{K} L^\alpha X^l W_\alpha
\]

where \( L^\alpha X^l \in \mathbb{R}^{V \times f_l} \), \( W \in \mathbb{R}^{f_l \times f_l} \), usually \( |V| >> f_1 > f_2 \) due to two facts:

- \( |V| \) is very large. \( L^\alpha X^l \) is usually sparse.
- In higher layers of DNNs, the feature dimension is usually decreasing, i.e. \( f_1 > f_2 \).

According to lemma on matrix multiplication:

**Lemma 1.** For matrix multiplication \( B = AW \), \( \text{Rank}(B) \leq \min \{ \text{Rank}(A), \text{Rank}(W) \} \)

The rank for GCN output feature matrix is bounded:

\[
\text{Rank}(G_W(X^l; A)) \leq \min \{ \text{Rank}_{W_a}(L^\alpha X^l), \text{Rank}_{f_1}(L^\alpha X^l), \text{Rank}_{f_1}(W_a), \text{Rank}_{f_1}(W_a) \}
\]

where \( \text{Rank}_{f_1}(W_a) \) is the rank on \( f_2 \) mode of matrix \( W_a \). Increasing the \( \text{Rank}(W_a) \) on both modes \( (f_1 \text{ and } f_2) \) will lift the upper bound of the output rank. It’s known that co-adaptation problem [8] limits the generality and transferability of DNNs. Initializing and freezing the covariance matrix along input and output dimension to \( I_f \) and \( O_f \), will induce a high rank matrix \( W_a \), which lifts the upper bound of rank of output features. The inter-neuron dependency is smaller for a high rank output feature matrix, so that the co-adaptation problem is alleviated and model generality is increased.

**Multi-modality fusion**

The final layer is the modality fusion layer, in order to aggregate features from different modalities and output a prediction result. For one-step spatiotemporal prediction problem, the output shape is \( \mathbb{R}^{V \times 1} \). The design of modality fusion is straightforward. First, we make sure the last MRGCN layer reduces the feature dimension to 1. Then, the modality fusion layer is designed as an modality-wise average:

\[
O^{l+1} = \frac{1}{|M|} \sum_{j=1}^{M} X^l_j, X^l_j \in \mathbb{R}^{V \times 1}
\]

Where \( M \) is the number of modalities.

**Training algorithm**

We combine all loss functions and summarize it for the entire network:

\[
J_W(X, Y) = \sum_{s \in S} f(W(x_s), y_s) + \alpha_{low} \sum_{l \in L_{low}} \| f_W(x) - y_s \|_2^2
\]

\[
+ \alpha_{low} \sum_{l \in L_{low}} (\alpha \sum_{i,j=1,\ldots,|M|} ||W_i^j|| + \sum_{i,j=1,\ldots,|M|} ||W_i^j||)
\]

\[
+ \alpha_{high} \sum_{l \in L_{high}} \frac{1}{2} (\text{vec}(W^l)^T (\Sigma^l)^{-1} \text{vec}(W^l))
\]

where the \( J_0 \) term is the prediction loss of the model. In this work, we use the root mean squared error (RMSE) to measure distance between the predicted value and true value. In stacked MGCNs, we set \( 1, 2, \ldots, l_k \)-th layers to \( L_{low} \) and use GCNN to construct graph interactions. The remaining layers \( l_h, l_h + 1, \ldots \) are set to learn multilinear relationships by MRGCN. The \( f_1 \) terms are the GGCN regularizer for each lower layer. The \( f_2 \) terms are the relationship regularizer for MRGCN in the higher layers. \( \alpha_{low} \) and \( \alpha_{high} \) are the trade-off parameters for regularizers.

**Algorithm 1** Training algorithm for the entire network, including GGCN and MRGCN is shown below.

**4 EXPERIMENTS**

In this section, we compare our graph interaction techniques with state-of-the-art baselines on region-level demand forecasting for ride-hailing service.

**Dataset**

We conduct our experiments on two real-world large scale ride-hailing datasets collected in two cities: City A and City B. Both of the datasets were collected in main city zone in 2017. We split data to training set (Mar 1st to Jul 31st, 2017), validation set (Aug 1st to Oct 31th, 2017) and test set (Nov 1st to Dec 31st, 2017). The POI data used for \( A_S \) contains 13 primary categories, including business, building, residential building, etc. The road network data used for \( A_C \) is extracted from railway, highway and subway dataset from OpenStreetMap [7].
Experiment setting

The ride-hailing forecasting problem is a one-step spatiotemporal prediction problem to learn predictor $f : \mathbb{R}^{V \times T} \rightarrow \mathbb{R}^{V \times |M|}$. According to previous works [6][11][29], we set $T$ to 5. Physically, it means to predict the ride-hailing demand in the next time interval using the most recent three ones (closeness), the one in the same time yesterday (period) and the one in the same time last week (trend)[30]. $V$ is the set of regions acquired by partitioning the main city zone to $1km \times 1km$ rectangular grids. Under this setting, there are totally 1296 regions in City A and 896 regions in City B. We set 30 minutes as the time interval for both training data and test data. Each entry in the spatiotemporal tensor represents the number of ride-hailing demand of a certain region in 30 minutes.

We propose a 4-layer MGCN, where the first two layers are GGCN and the last two layers are MRGCN. The output dimensions for these layers are set to 32,64,32,1. For all graph convolution operations, the max chebyshev polynomial $K$ is set to 4. In GGCN, the tunable $\epsilon$ is set to 0.1 to maintain intra-modality properties. In MRGCN, the trade-off parameter $\alpha$ is set to $1e-6$. We monitor RMSE on the validation set with early stopping. The regularizers $\ell_{1}$ and $\alpha_{high}$ are both set to $1e-4$. The neural network is implemented using tensorflow [1] and optimized using adam optimizer [12] with the learning rate as 5e-4 and the batch size as 32. All experiments are conducted on an environment with 10GB RAM and 9GB GPU memory of Tesla P40.

Performance comparison

| Method      | Lower layers | Higher layers | RMSE in City A | RMSE in City B |
|-------------|--------------|---------------|----------------|----------------|
| STMGCN      |              |               | 10.78          | 8.30           |
| MGCN        |              |               | 11.82          | 8.64           |
| GGCN        |              |               | 9.51           | 8.18           |
| MRGCN$_{\Sigma}$ |              |               | 9.68           | 8.30           |
| GGCN        | Share weight |               | 9.59           | 8.13           |
| GGCN        | DAN          |               | 9.48           | 8.02           |
| GGCN        | MRGCN$_{\Sigma}$ |               | 9.47           | 7.92           |
| GGCN        | MRGCN$_{\Sigma}$ |               | 9.31           | 7.88           |

Table 2: Experiment performance in City A and City B. The proposed approach achieves best result among all methods.

Table 3: Number of epochs required to converge to optima or benchmark. Multi-task-based method reduce training time by at least 50%. The experiment is done in City A dataset.

| Method      | Lower layers | Higher layers | Epoch of converge | Epoch to break 10.78 |
|-------------|--------------|---------------|--------------------|----------------------|
| STMGCN      |              |               | 115                | 115                  |
| MGCN        |              |               | 110                | -                    |
| GGCN        |              |               | 130                | 55                   |
| MRGCN$_{\Sigma}$ |              |               | 95                 | 32                   |
| GGCN        | Share weight |               | 78                 | 32                   |
| GGCN        | DAN          |               | 72                 | 24                   |
| GGCN        | MRGCN$_{\Sigma}$ |               | 51                 | 25                   |
| GGCN        | MRGCN$_{\Sigma}$ |               | 82                 | 27                   |

Multi-Modal Graph Interaction for Multi-Graph Convolution Network in Urban Spatiotemporal Forecasting

Conference’17, July 2017, Washington, DC, USA

- **MGCN**: Use one separate GCN to learn prediction task in each modality. There is no graph interaction among modalities.
- **STMGCN**[6]: Use RNN-based model to extract temporal features ahead of GGCN.
- **Share weight**: A common technique in multi-task learning. The GCN weight is shared across modalities in each layer.
- **Domain adaptation network (DAN)**[16]: Minimizing modality divergence by minimizing cross-modality feature divergence. The divergence used is mean maximum discrepancy (MMD).
- **MRGCN$_{\Sigma}$**: The proposed multi-linear relationship GCN with all four covariance matrices updated.
- **MRGCN$_{\Sigma}$**: Proposed method to freeze covariance matrices for input and output coordinates.

All proposed methods above are 4-layer MGCNs, with similar hidden feature sizes and same training configurations (learning rate, batch size, etc). We evaluate the model performance according to the prediction error (RMSE) on the test set. The epoch of converge shown in table 4 measures the time consumption for each model to reach its optima. Different models converge to different optima. Achieving a lower error usually costs longer training time. We set the benchmark to 10.78 in City A, which is the performance of baseline [6] on the same dataset.

The experiments shows following facts. Firstly, according to performance of GGCN, it improves the prediction accuracy for MGCN by invoking more complexity in spatial feature extraction on graphs. With the help of intra-modality transformations, spatial feature extraction is more complete and the model is more expressive. The performance improvement by GGCN is even more significant than incorporating an RNN-based temporal feature extraction process (STMGCN). However, with the increment of the parameter size, the model is more prone to overfitting and requires longer training time.

Secondly, MRGCN also improves model performance. Compared with GGCN, the influence to prediction error is slightly inferior. There is no significant difference in model capacity and model structure between MRGCN and MGCN. We infer that multi-linear relationship approach improves prediction performance by improving model generality, so that $\text{MRGCN}_\Sigma$ is less prone to overfit to the local fluctuations in training set and overcomes the gap between training set and test set. Multi-task learning based approaches, including share weight, DAN and MRGCN all shorten the model training time. Among these approaches, share weight method reduces model complexity by a factor of $O(|M|)$, which brings down the prediction performance. The performance of MRGCN and DAN are almost the same.

Thirdly, we show that freezing input and output coordinates in MGCN is effective. Compared with $\text{MRGCN}_\Sigma$, $\text{MRGCN}_\Sigma$ decreases the prediction error. This validates our assumption that
freezing the covariance for input and output dimension on the weight tensor may induce higher independency among neurons, which alleviates the co-adaptation problem, thus improves model generality.

Training speed is another important factor to evaluate machine learning models. Table 4 shows the training time required to achieve the optimal performance of each model. We use the grid search to determine the minimum training length of each model. Given a larger training set than this, the model can’t converge to a significantly lower validation error. Compared with the baseline, the proposed method reduces the amount of training set and the length of training by approximately 50%. Among all tested approaches, MRGCN achieves the lowest prediction error on average and on test data after the 4th week. This is an important feature for industrial use. The life cycle for a more generalized model is longer, which reduces the frequency for model update.

**Model generality**

| Min training length | STMGCN | GGCN | MRGCN | GGCN+MRGCN |
|---------------------|--------|------|-------|------------|
| Training time       | 5 months | 3 months | 3 months | 3 months |

Table 4: Training speed for each model to achieve best performance in ride-hailing demand forecasting task.

Figure 5 shows the generalization ability for different models, which validates above arguments in detail. The data relative divergence (blue bar) is computed as the Kullback Leibler divergence[13] between temporal pattern of the last week in training set, and temporal patterns of each week in test set. We discovered that the gap between training set and test set is accumulative. This indicates that the test data will become more and more divergent from training data with time shifting. Models are expected to be more general to overcome this phenomenon. According to prediction error by weeks, the prediction error for STMGCN keeps increasing as the test data becomes more divergent. We believe this phenomenon is not caused by model capability, but model generality. For methods including GGCN and MRGCN, the model performance is less influenced by this generalization gap. There is no difference between the model capacity of STMGCN (MGCN) and MRGCN. The network architecture and connectivity are almost the same. This shows that MRGCN has better generalization ability to avoid overfitting to training set.

Figure 6 shows the feature dimension-wise covariance for different models. The feature covariance is calculated as negative logarithm of the L2-norm of the covariance matrix of latent features along the feature mode. A higher value represents higher feature independence.

Figure 6: Feature dimension-wise covariance for different models. It’s calculated as the negative logarithm of the L2-norm of the covariance matrix of latent features along the feature mode. A higher value indicates higher feature independence.

**Modality relationship**

MRGCN learns explainable relationships between modalities by maintaining a modality-wise covariance matrix. In this part, we first show that all modalities are helpful to the learning task. Then, we will explore the relationship between the modality-wise relationship learnt from optimization and relationship between graphs.

Figure 7 is the Hinton diagram showing the modality-wise relationships for the 3rd and 4th layers in GGCN + MRGCN. N, P, R represent modality for Neighborhood AN, POI similarity AS and road connectivity AR. Similar to the interpretation by [17], we could draw several conclusions. Firstly, most of the tasks are
positively correlated (green), implying that all modalities could reinforce the learning of others. This conclusion reaches a consensus with ablation study of [6] in table 5.

| Removed component | RMSE |
|-------------------|------|
| Neighborhood      | 11.47|
| POI similarity    | 11.42|
| Road connectivity | 11.69|
| ST-MGNCN          | 10.78|

Table 5: Ablation study for ST-MGNCN. Removing any one modality will result in great damage to the prediction accuracy.

Secondly, we discover that the relationship between N and R is weak and random. These two tasks are seemingly related. Compared with that, the relationship R-P and N-P are stable and robust. We try to explain this phenomenon by comparing the graphs $A_N$, $A_S$ and $A_C$.

$$A_N \sim 1.3e^{-3} \quad A_S \sim 1.4 \quad A_C \sim 1.4e^{-3}$$

Table 6: The density for each graph. The graphs are undirected. Density is calculated as $2|E|/|V||V|-1|

Table 7: Two measurements to show similarity between different graphs. F-measure considers matched and unmatched edges proportional to graph size. Edit distance measures difference between two edge sets.

Table 6 shows the density for each graph, which measures the connectivity of graph in each modality. According to graph definition, $A_S$ is defined as POI similarity between any region pairs, which induces a dense adjacency matrix. $A_N$ and $A_C$ are sparse. We measure the graph similarity by F-measure and edit distance in table 7. According to graph definition, edges in $A_N$ are all removed from $A_C$, that the edge set $E_N \bigcap E_C = \emptyset$. From the view of graph connectivity, the prediction task on these modalities are hardly related. The relationship $A_S \sim A_C$ and relationship $A_S - A_N$ are quite similar due to that $A_S$ is dense. The analysis above helps to understand figure 7. The relationship between neighborhood (N) and road connectivity (R) is quite random, due to the inherent independence between these two modalities. MRGCN learns similar modality relationships for similar graph-pairs. The relationship N-P and R-P are maintained to be similar in both layers.

5 CONCLUSION AND FUTURE WORK

In this work, we propose two graph interaction techniques for multi-modal multi graph convolution networks. We use GGCN in lower layers to complete graph connectivity for better spatial feature extraction by graph convolution networks. In higher layers, we use MRGCN to learn robust modality relationships. MRGCN alleviates the co-adaptation problem by lifting the upper bound for feature dependency, thus improves the model generality. The experiment on ride-hailing demand prediction shows that our proposed model outperforms baselines in effectiveness, efficiency and robustness. For the future work, we plan to investigate the following aspects: (1) evaluate the model with other spatial temporal prediction tasks and other region-wise relationships; (2) explore the impact of sparse and dense graphs on this framework;

REFERENCES

[1] Martin Abadi et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI ’16).

[2] Pradeep K. Atrey, M Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S. Kankanhalli. 2010. Multimodal fusion for multimedia analysis: a survey. Multimedia systems 16, 6 (2010), 345–379.

[3] Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2018. Multimodal machine learning: A survey and taxonomy. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018).

[4] Di Chai, Leye Wang, and Qiang Yang. 2018. Bike Flow Prediction with Multi-Graph Convolutional Networks. SIGSPATIAL (2018).

[5] Michael Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In Advances in Neural Information Processing Systems. 3844–3852.

[6] X. Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. 2019. Spatiotemporal Multi-Graph Convolution Network for Ride-hailing Demand Forecasting. In 2019 AAAI Conference on Artificial Intelligence (AAAI’19).

[7] Mordechai Haklay and Patrick Weber. 2008. Openstreetmap: User-generated street maps. IEEE Pervasive Computing 7, 4 (2008), 12–18.

[8] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012).

[9] Micah Hodos, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. Journal of Artificial Intelligence Research 47 (2013), 853–899.

[10] J. Ke, H. Yang, H. Zheng, X. Chen, Y. Jia, P. Gong, and J. Ye. 2018. Hexagon-Based Convolutional Neural Network for Supply-Demand Forecasting of Ride-Sourcing Services. IEEE Transactions on Intelligent Transportation Systems (2018), 1–14. https://doi.org/10.1109/TITS.2018.2882961

[11] Jintao Ke, Hongyu Zheng, Hai Yang, and Xiquan Michael Chen. 2017. Short-term forecasting of passenger demand under on-demand ride services: A spatiotemporal deep learning approach. Transportation Research Part C: Emerging Technologies 85 (2017), 591–608.
[12] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In International Conference on Learning Representations (ICLR '15).

[13] Solomon Kullback and Richard A. Leibler. 1951. On information and sufficiency. The annals of mathematical statistics 22, 1 (1951), 79–86.

[14] 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 '18).

[15] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In International Conference on Learning Representations (ICLR '18).

[16] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. 2015. Learning transferable features with deep adaptation networks. arXiv preprint arXiv:1502.02791 (2015).

[17] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and S Yu Philip. 2017. Learning multiple tasks with multilinear relationship networks. In Advances in Neural Information Processing Systems. 1594–1603.

[18] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Advances in neural information processing systems. 802–810.

[19] Xingjian Shi, Zhihao Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. 2017. Deep learning for precipitation nowcasting: A benchmark and a new model. In Advances in Neural Information Processing Systems. 5617–5627.

[20] Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) (1996), 267–288.

[21] Yongxin Tong, Yaqiang Chen, Zimu Zhou, Lei Chen, Jie Wang, Qiang Yang, Jieping Ye, and Weifeng Lv. 2017. The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 1653–1662.

[22] Ying Wei, Yu Zheng, and Qiang Yang. 2016. Transfer Knowledge between Cities. In KDD. 1905–1914.

[23] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, Yanwei Yu, and Zhenhui Li. 2018. Modeling Spatial-Temporal Dynamics for Traffic Prediction. arXiv preprint arXiv:1803.01254 (2018).

[24] Huaxiu Yao, Pei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. 2018. Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction. In 2018 AAAI Conference on Artificial Intelligence (AAAI'18).

[25] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?. In Advances in neural information processing systems. 3320–3328.

[26] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In IJCAI'18.

[27] Ming Yuan and Yi Lin. 2006. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68, 1 (2006), 49–67.

[28] Ben P. Yulius, Moise H. Goldstein, and Terrence J. Sejnowski. 1989. Integration of acoustic and visual speech signals using neural networks. IEEE Communications Magazine 27, 11 (1989), 65–71.

[29] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In AAAI. 1655–1661.

[30] Junbo Zhang, Yu Zheng, Dekang Qi, Ruyuan Li, and Xiwen Yi. 2016. DNN-based prediction model for spatio-temporal data. In SKS/PATAL. 92.

[31] Yu Zhang and Qiang Yang. 2017. A survey on multi-task learning. arXiv preprint arXiv:1707.08114 (2017).

[32] Yu Zhang and Dit-Yan Yeung. 2012. A convex formulation for learning task relationships in multi-task learning. arXiv preprint arXiv:1203.3536 (2012).