C02029: Doctor of Philosophy *Subject NO.: 32903*

August 2020

Network-wide Spatio-Temporal Predictive Learning for the Intelligent Transportation System.

Yongshun Gong

School of Electrical and Data Engineering
Faculty of Engg. & IT
University of Technology Sydney
NSW - 2007, Australia

Network-wide Spatio-Temporal Predictive Learning for the Intelligent Transportation System.

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

by

Yongshun Gong

to

School of Electrical and Data Engineering
Faculty of Engineering and Information Technology

University of Technology Sydney
NSW - 2007, Australia

August 2020

AUTHOR'S DECLARATION

I, *Yongshun Gong* declare that this thesis, submitted in partial fulfillment of the requirements for the award of Doctor of Philosophy, in the *School of Electrical and Data Engineering*, *Faculty of Engineering and Information Technology* at the University of Technology Sydney, Australia, is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution. This research is supported by the Australian Government Research Training Program.

Production Note:

SIGNATURE: Signature removed prior to publication.

[Yongshun Gong]

DATE: 05th August, 2020

PLACE: Sydney, Australia

Acknowledgments

My deepest gratitude goes first and foremost to Prof. Jian Zhang, my principal supervisor, for his continuous support and guidance to my Ph.D. study. He has walked me through all the stages of the writing of this thesis, which greatly improved my understanding of academic writing and taught me a large amount of specific research skills. His consistent and illuminating instruction have helped me to grow as a research scientist.

Second, I would like to express my heartfelt gratitude to my co-supervisor Prof. Qiang Wu and collaborators: Dr. Wei Liu, Dr. Jinfeng Yi, Dr. Bei Chen, and Dr. Yu Zheng, for not only their constant encouragement and guidance but also for the thought-provoking questions, which have helped me to broaden my research horizon and develop the fundamental and essential academic competence.

I am also greatly indebted to all my labmates in the Global Big Data Technologies Centre: Zhibin Li, Lu Zhang, Junjie Zhang, Xiaoshui Huang, Muming Zhao, Huaxi Huang, Anan Du and Lingxiang Yao, for the weekly discussions around our research problems and enthusiastic atmosphere we worked together. I am very appreciate their encouragement and support in the past three years.

Last but not least, my heartfelt gratitude would go to my beloved family for their selfless supports and great confidence in me during my Ph.D. study.

List of Publications

Journals:

1. **Yongshun Gong**, Zhibin Li, Jian Zhang, Wei Liu, Yu Zheng. Online Spatio-temporal Crowd Flow Distribution Prediction for Complex Metro System. *in* IEEE Transactions Knowledge and Data Engineering (TKDE), 2020.

2. Dong, Xiangjun, **Yongshun Gong***, and Longbing Cao. "e-RNSP: An efficient method for mining repetition negative sequential patterns." *in* IEEE transactions on cybernetics (TCYB), 2020: 2084-2096.

3. Dong, Xiangjun, **Yongshun Gong**, and Longbing Cao. "F-NSP+: A fast negative sequential patterns mining method with self-adaptive data storage." *in* Pattern Recognition (PR), (84) 2018: 13-27.

4. Xinming Gao, **Yongshun Gong**, Tiantian Xu, Jinhu Lv, etc.Towards to a Better Structure and Looser Constraint to Mine Negative Sequential Patterns. *in* IEEE transactions on Neural Networks and Learning Systems (TNNLS), 2020, accepted, Xinming Gao and Yongshun Gong contributed equally.

Conferences:

5. **Yongshun Gong**, Zhibin Li, Jian Zhang, Wei Liu, Bei Chen, Xiangjun Dong. A Spatial Missing Value Imputation Method for Multi-view Urban Statistical Data. *in* Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI20). pp. 1310-1316.

6. **Yongshun Gong**, Zhibin Li, Jian Zhang, Wei Liu, Jinfeng Yi. Potential Passenger Flow Prediction: A Novel Study for Urban Transportation Development. *in* Proceedings of the AAAI Conference on Artificial Intelligence (AAAI20). pp. 4020-4027.

7. **Yongshun Gong**, Zhibin Li, Jian Zhang, Wei Liu, Yu Zheng, Christina Kirsch.

Network-wide Crowd Flow Prediction of Sydney Trains via customized Online Non-negative Matrix Factorization. *in* Proceedings of the Conference on Information and Knowledge Management (CIKM18), pp. 1243-1252.

8. Zhibin Li, Jian Zhang, Qiang Wu, **Yongshun Gong**, Jinfeng Yi, Christina Kirsch. Sample Adaptive Multiple Kernel Learning for Failure Prediction of Railway Points. *in* Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (SIGKDD19), pp. 2848-2856.

9. Zhibin Li, Jian Zhang, **Yongshun Gong**, Yazhou Yao, Qiang Wu. Field-wise Learning for Multi-field Categorical Data, NeurIPS-2020, accepted.

10. Lu Zhang, Jingsong Xu, Jian Zhang, **Yongshun Gong**. Information Enhancement for Travelogues via a Hybrid Clustering Model. *in* Proceedings of the Digital Image Computing: Techniques and Applications (DICTA18), pp. 1-8.

Under Review:

1. **Yongshun Gong**, Zhibin Li, Jian Zhang, Wei Liu, Yu Zheng. Missing Value Imputation for Multi-view Urban Statistical Data via Spatial Correlation Learning. Submitted to IEEE Transactions Knowledge and Data Engineering (TKDE).

2. **Yongshun Gong**, Jinfeng Yi, Dong-Dong Chen, etc. Inferring the Importance of Items Appearance: A Step towards the Screenless Retailing. Submitted to WWW-2021.

3. **Yongshun Gong**, Bei Chen, Jianguang Lou. An Exploratory Study on the Linguistic Structure in Text-to-SQL. Submitted to ACL-2021.

4. Zhibin Li, **Yongshun Gong**, Jian Zhang, Yazhou Yao, Qiang Wu. Missingness-pattern-adaptive Learning with Incomplete Data, Submitted to ICML-2021.

5. Ping Qiu, **Yongshun Gong**, Xiangjun Dong, Longbing Cao, Chengqi Zhang. An Efficient Method for Mining Negative Sequential Patterns Under Loose Constraints. Submitted to IEEE Trans. Neural Networks and Learning Systems (TNNLS).

6. Lu Zhang, Jingsong Xu, **Yongshun Gong**, Jian Zhang. Image and Text Fusion for Travel Information Enhancement via Multi-View Embeddings. Submit to IEEE Transactions on Multimedia (TMM).

Table of Contents

List of Figures

Figure **Page**

LIST OF TABLES

TABLE

Abstract

LLarge volumes of spatio-temporal data are increasingly collected and benefited to diverse domains, including transportation, urban optimization, community detection, climate science, etc. How to feed these large-scale data into a network-wide prediction model for the intelligent transportation system is a promising problem. Currently, even though a number of traffic prediction models have been proposed to enhance the travel services and improve operational performance of transit authorities, limited methods can be applied to forecast the network-wide traffic conditions afterward.

This thesis focuses on three problems in our predictive task. Firstly, the spatio-temporal data usually suffers from the missing data problem. Those missing values hide the useful information that may result in a distorted data analysis. In Chapter 3, a spatial missing data imputation method is proposed for multi-view urban statistical data. To address this problem, our method exploits an improved spatial multi-kernel clustering approach to guiding the imputation process cooperating with an adaptive-weight non-negative matrix factorization strategy. Secondly, in the crowd flow prediction, most existing techniques focus solely on forecasting entrance and exit flows of metro stations that do not provide enough useful knowledge for traffic management. In practical applications, managers desperately want to solve the problem of getting the potential passenger distributions to help authorities improve transport services, termed as crowd flow distribution (CFD) forecasts. Therefore, to improve the quality of transportation services, three spatiotemporal models are designed in Chapter 4 to effectively address the network-wide CFD prediction problem based on the online latent space (OLS) strategy. Our models take into account the various trending patterns and climate influences, as well as the inherent similarities among different stations that are able to predict both CFD and entrance and exit flows precisely. Lastly, with the development of urbanization, a real-world demand from transportation managers is to construct a new metro station in one city area that never planned before. Authorities are interested in the picture of the future volume of commuters before constructing a new station, and estimate how it would affect other areas. In this thesis, the specific problem is termed as potential passenger flow (PPF) prediction. Chapter 5 proposes a multi-view localized correlation learning method to provide a solution for the PPF prediction that can learn localized correlations via a multi-view learning process.

CHAPTER 1

INTRODUCTION

1.1 Background

Urbanization's rapid progress has modernized many people's lives but also engendered huge issues in transportation system, such as complex route planning, traffic congestion, traffic safety, etc. Intelligent transportation systems (ITS), being the important part of the urban computing, has gained extensive attentions to address such issue to improve the quality of living conditions. Nowadays, large-scale computing infrastructures and sensing technologies have produced a large volumes of spatio-temporal data in urban spaces ((e.g., traffic flow data, statistical data, and geographical data) [103]. These spatio-temporal data imply rich knowledge about a city that can help address above challenges in transportation system when used effectively and correctly. For instance, we can predict the crowd flow in a city's subway network through analyzing the city-wide commuting data. This investigation betters the formulation of city's transportation planning [105]. Another example is to forecast the areas' potential crowd flow. Transportation managers are interested in the picture of the future volume of commuters before building a new station, and estimate how would it affect other regions. For this specific issue, a potential solution is to mine correlations between regional statistical data and other spatio-temporal data sources, such as traffic flow and points of interest (POIs) [104]. The goal of this thesis is to study three main problems in the network-wide spatiotemporal predictive learning task for the intelligent transportation system.

First, to utilize the spatio-temporal data effectively, one big issue is the missing data problem. For the traffic data, there are numerous sensors collecting traffic data in road, but they distribute unevenly and probably change over time. In the extreme case, no data is generated in some roads during a time interval. Besides, malfunctions of sensors and human factors also lead to the problem of missing data. As a result, traffic data can be of high sparsity, which makes missing data completion to be an important step in traffic prediction. For another urban spatial data, e.g., urban statistical data, in some places, they are hard to be entirely acquired due to document defacement, error recordings, and statistician misplay. Such missing data hide useful information which may cause distorted results for further urban development analysis. This thesis will focus on addressing the missing data problem in urban statistical data because there exists several specified challenges in this domain, and to the best of our knowledge, none of existing methods can solve this problem well. To date, a number of missing data imputation approaches can be applied in urban statistical data, e.g., mean-filling (MF), k-nearest-neighbor (KNN) filling [57], and collaborative filtering based methods [62]. Most of them, however, have been proposed to focus on the single view problem. Besides, although several spatiotemporal methods can infer the missing information based on the knowledge from both spatial and temporal domains [13, 92, 106], they do not perform well when the temporal information missed. This problem will be discussed and resolved in Chapter 3.

Second, similar as the traffic prediction problem, forecasting crowd flows in a city trains network is strategically significant because of the benefits it brings to many metro management and urban optimization services, such as congestion avoidance, route scheduling, public safety, and so forth [55, 103]. It is very important for public safety: for instance, streamed people caused a chaotic crowd stampede at the Falls Festival in Lorne on Victoria's south-west coast, leaved up to 80 people injured; and 36 people died in a catastrophic stampede at the 2015 New Year's Eve celebrations in Shanghai [97]. An effective crowd warning and prediction system can effectively prevent people from such real tragedies by utilizing emergency mechanisms.

Thanks to the transportation smart card ticketing system for travel on public transport, a large amount of transactional data is now available that contains very detailed information. Based on these useful data, a number of applicable passenger flow prediction models have been proposed to enhance the metro services and improve operational performance of transit authorities [59, 100]. The existing techniques for addressing crowd flow prediction problems are mainly based on regression strategies like auto-regressive integrated moving averages (ARIMA) [84] or Gaussian processes (GP) [107]. Other strategies, such as neural networks [37, 39, 83], probability trees [36] and wavelet-SVM [70] have also been proposed as solutions to passenger flow prediction problems. However, to the best of our knowledge, none of these techniques can be used directly to predict crowd flows across an entire train network. The details of the network-wide crowd flow prediction problem are shown in Chapter 4.

Third, with the growth of intelligent transportation systems, passenger flow prediction models concentrate on discovering the volume of crowds and mobility patterns that best serve people's daily life [58, 98]. Recent advances in passenger flow prediction are focusing mainly on next time interval flow conditions with time evolves [22, 68]. If a brand-new metro station is inserted into the original metro network, existing predictors have to collect a large amount of latest transactional data to ensure normal operation. However, a real-world requirement from transportation authorities is that they want to obtain the potential passenger flows (PPF) of a planned city area in advance (i.e., before constructing a station in this area). It is significant for the urban traffic development and transportation management, as it can provide insights for the site selection of stations and analysis of passenger movement patterns, as well as give the potential crowd warning. To date, limited studies considered the OD passenger flow prediction problem [22, 81], and none of existing techniques focus on forecasting PPF across the entire city. It is a novel problem connecting to the urban development and has been discussed in 5.

1.2 Research Challenges

1.2.1 Spatial Missing Data Imputation

In this thesis, the study has been explored in the missing-data imputation problem for the Australian Bureau of Statistics (ABS) data, which has some unique challenges:

• *Missing temporal information.* In the real-world data from ABS, almost all the missing values in the current year were also missing in the past years, which may be caused by the region restriction and complicated human-made errors. This violates the basic assumption of matrix completion [7] that the unobserved entries are sampled uniformly at random. Thus matrix completion-based approaches may not work in this case.

• *Multi-view problem.* The complicated underlying interactions suggest that simply recovering the missing information without considering the correlations among attributes

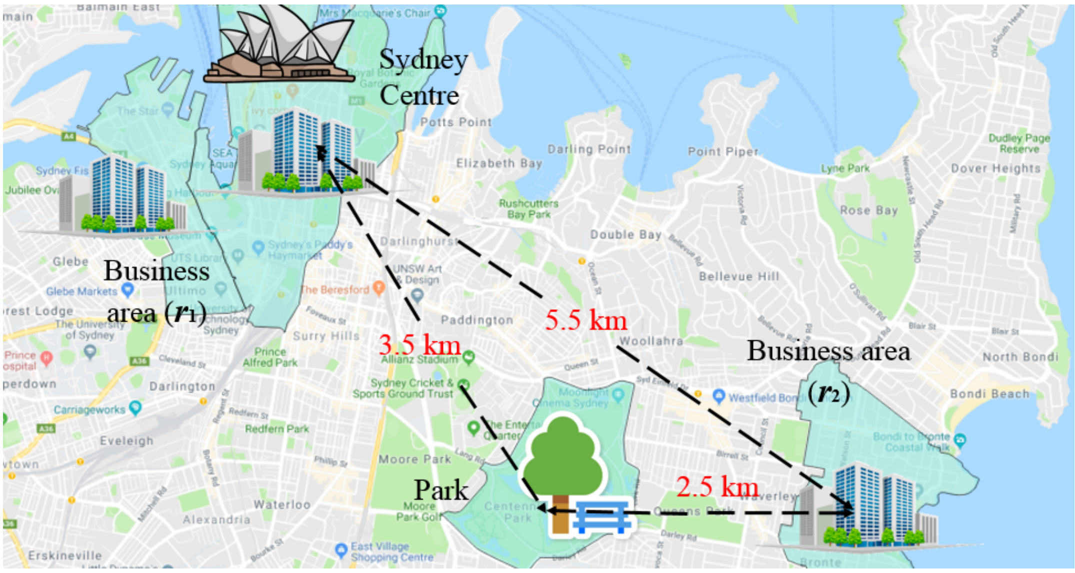


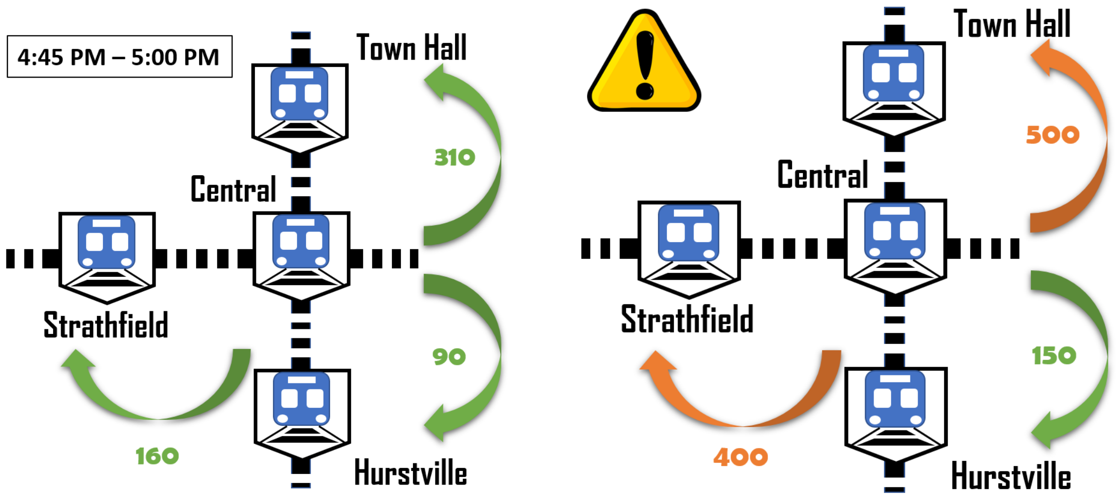
Figure 1.1: Regional similarity: the property of r_1 is similar to the 'Sydney centre' because they are neighboring each other. Although r_2 is closer to the park in terms of the physical distance, the attributes of r_2 are more analogous to 'Sydney centre' than the park because they have a similar functional property (business centre).

and multi-modes will end up with a poor performance. For example, the economy view has strong correlations with the income and population views, so that a high-quality economy in a region usually goes along with a better income and a larger population; and a low-level economy in a region has a high probability of being connected with a lower income and a smaller population.

- *Spatial correlation mining problem.* As illustrated in Figure 1.1, the statistical data focusing on fine-grained regions may change over locations significantly and non-linearly. Therefore, to properly recover the missing information of statistical data, the regional similarities need to be considered.

1.2.2 Crowd Flow Distribution (CFD) Prediction

In many real-world applications, concentrating solely on entrance and exit flows does not provide adequate information, managers also need to know potential passenger distributions, i.e., CFD forecasts. Figure 1.2 illustrates an example CFD prediction. Figure 4.1(a) presents a predicted snapshot. The model makes a forecast that there are 560 passengers departure from Central station between 4:45 PM and 5:00 PM. Among them, 310 passengers will arrive at Town Hall, 160 at Strathfield, and 90 at Hurstville, respectively. Through obtaining the CFD forecasts among all metro stations, transport managers can timely forecast irregular flow patterns and make a global regulation to maintain the normal train scheduled and make a warning for crowd evacuation. Figure 4.1(b) illustrates this situation that when an irregular entrance flow appears in the Central station, the congestion warning will be transmitted to the all possibly affected stations (Strathfield and Town Hall). A CFD model can illustrate the crowd flows among all these stations, which is significant for passenger route planning, train scheduling, and crowd warning systems. These models could be especially useful for predicting passenger flows during irregular events, such as train faults, emergencies, and public events, where passenger flows may suddenly surge over a short time span. With a strong CFD model, a transport administrator could forecast abnormal flow patterns and plan crowd evacuations for all affected stations to ensure public safety and/or maintain the normal train scheduled, as shown in Figure 1.2 (b).



(a) 560 passengers enter at Central between 4:45 to 5:15 pm, and the distribution of this entrance flows.

(b) A congestion warning for all possibly affected stations when occurring non-recurrent events.

Figure 1.2: An example of the crowd flow distribution.

To date, limited techniques can be used directly to address the network-wide CFD prediction problem. Regression-based methods like Gaussian processes (GP) [107] and auto-regressive integrated moving averages (ARIMA) [84] are proposed to forecast entrance and exit crowd flows. While other approaches, such as wavelet-SVM [70] and probability trees [36] have successfully designed address the classical crowd flow prediction problem, they are hard to implement into the entire metro network. Even though deep neural networks [83, 96, 97], are able to fix the network-wide crowd flow prediction problem, they are sensitive to parameters and incomplete inputs, and require large training data that are not in line with our task. To summarize, our CFD prediction problem faces with three intrinsic challenges here:

• *High computational complexity.* The specific CFD prediction problem requires getting all potential flows across entire metro stations, which calculates the entrance/exit flows and CFD simultaneously. Most advanced models like [10, 12, 55, 70, 83], are already computationally expensive even on a few metro lines. Meanwhile, they require repeated large off-line training processes that are difficult to be applied in the online system and network-wide problem.

• *Dynamic complexity.* The crowd flow changes dynamically which is influenced by complicating factors, such as time, station similarity and climate conditions.

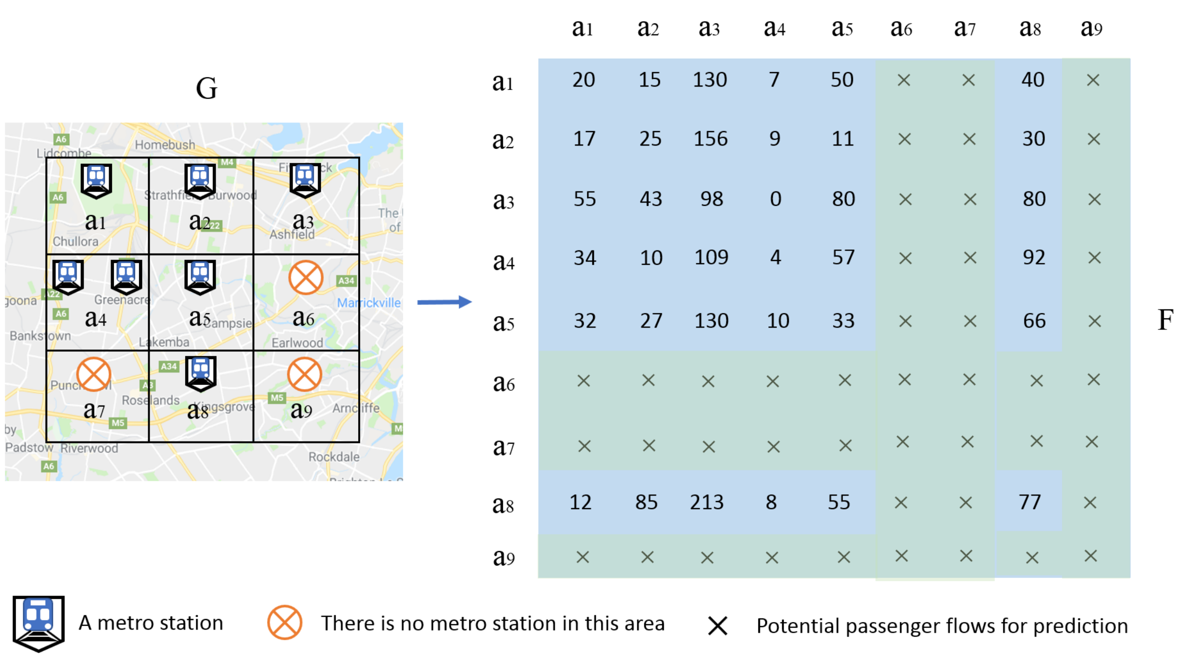
• *Real-time delayed data collection.* Considering the online system, when we focus on entrance CFD prediction, there is a travel time gap between a passenger enters a station and exits another. These time gaps lead to the online system cannot collect complete data because there are a large number of passengers still on their journeys. In this situation, most city-wide traffic flow prediction methods, such as [23, 96–98], fail to solve our problem because they require the complete data in training and testing processes.

1.2.3 Potential Crowd Flow Prediction

One of the novel problems in transportation system is named potential crowd flow prediction (PPF). In this problem, we aim to discover the latent connections among diverse domains, i.e., utilize correlations between commuting data and national statistical data to predict the potential passenger flows in some specified geographic regions where the subway (or city train) stations have not been built yet. Figure 1.3 illustrates this problem.

In the PPF prediction task, concentrating solely on the entrance and exit potential flows does not provide adequate information, authorities also desperately want to master the distribution of predicted PPF, i.e., forecast the number of potential passengers moving to different destinations. It is utmost important to find how will the new station affect other areas. For instance, Figure 1.3 illustrates an example of the PPF prediction problem. A city region is partitioned into nine areas[1], six of them have metro stations (termed as known areas), and three have not constructed yet (termed as target areas). The right part of Figure 1.3 presents an origin-destination (OD) matrix (each row point

[1] We use grids for clear and simple illustration, the real partition standard is explained in the Chapter 5.


G
a1
a2
a3
a4
a5
a6
a7
a8
a9
F
A metro station
There is no metro station in this area
Potential passenger flows for prediction

	a1	a2	a3	a4	a5	a6	a7	a8	a9
a1	20	15	130	7	50	×	×	40	×
a2	17	25	156	9	11	×	×	30	×
a3	55	43	98	0	80	×	×	80	×
a4	34	10	109	4	57	×	×	92	×
a5	32	27	130	10	33	×	×	66	×
a6	×	×	×	×	×	×	×	×	×
a7	×	×	×	×	×	×	×	×	×
a8	12	85	213	8	55	×	×	77	×
a9	×	×	×	×	×	×	×	×	×

Figure 1.3: The example of PPF prediction problem. PPF aims to forecast the passenger flows of target areas (e.g., a_6, a_7, a_9) across the entire city network.

is the origin area and column points are destinations), e.g., $F(a_1, a_3) = 130$ indicates that there are 130 passengers departure from a_1 and are going to the a_3. PPF task aims to make an accurate prediction for the target areas in one period (e.g., rush hours) that completes the crowd flows between them and known areas.

To date, limited studies considered the OD passenger flow prediction problem [22, 81], and to the best of our knowledge, none of existing techniques can forecast PPF across the entire city. It is a novel problem and a real urban developing demand that faces several major challenges:

- Considering the number of passenger flows and their final destinations simultaneously.

- Analogously to the cold-start problem in the recommender system [33], it is hard to infer the preference of a new user from the known data. In our problem, a new station in the target area can be similarly regarded as a new user.

- Since the PPF is a spatial-temporal mining problem, spatial and temporal information should be taken into account appropriately.

1.3 Research Contributions

After researching the above challenges, the author has developed corresponding solutions, which are presented in this thesis. These study contributions are showed as below:

- To handle the multi-view problem with spatial characteristic, we propose a Spatially related Multi-Kernel K-Means (S-MKKM) method to identify the underlying relationships among multiple views and capture the regional similarities. An adaptive-weight non-negative matrix factorization approach is proposed to leverage the information learned above to tackle the multi-view missing data imputation problem. Besides, the proposed method also takes the guidance from the single-view and the real geographic information with KNN strategy into consideration. A spatial multi-view missing data imputation method for urban statistical data based on non-negative matrix factorization is proposed, called SMV-NMF. SMV-NMF does not rely on the temporal information but achieves a great performance only using spatial information (Chapter 3).

- Our experiments on six real-world datasets verify the effectiveness of our method. All the empirical results show that the proposed method SMV-NMF outperforms all the other state-of-the-art approaches. Furthermore, SMV-NMF shows strong generalizability and can transfer the constructed model from one urban dataset to another well (Chapter 3).

- The network-wide CFD prediction problem is formulated as a graph network problem and propose a data-driven forecasting model, called OLS-AO, that combines current flow trends with historic guidance to address the three inherent challenges associated with network-wide crowd flow prediction. To further improve the effectiveness of the model in real-world situations, we designed another extended OLS model, called OLS-MR, that is able to adapt to sudden changes in crowd flows (Chapter 4).

- We proposed a dual track model, called OLS-TD, integrates both OLS-AO and OLS-MR to address a variety of challenging traffic scenarios. Our proposed models are compared with five available prediction methods in a set of intensive experiments on a large, real-world Opal Card dataset covering Sydney Trains. The experiments assess the models' effectiveness from four perspectives, including: CFD predictions across the entire network at different timestamps, CFD predictions at major stations, comparisons between weekdays and weekends, and comparisons between peak and non-peak times. The experimental results show that OLS-AO achieved the best results for the weekend tests, and OLS-TD proved to be more stable and effective for all weekday tests (Chapter 4).

- We devise a multi-view localized correlation learning model for the PPF prediction (MLC-PPF for short). To leverage the spatial information, our method first constructs a localized similarity matrix which associates with the real geographical neighbors and regional properties (e.g., business or residential regions). The intuition behind this strategy is from the First Law of Geography [75], i.e. *"Everything is related to everything else, but near things are more related than distant things"*. Second, a novel weighted correlation learning strategy is proposed. At last, to improve the prediction accuracy and well handle the cold-start challenge, we draw the side information from urban statistical data, where each area has a multi-view features to guide the learning process (Chapter 5).

- We show that our PPF method can be transferred to the classic cold-start problem in the recommender system. It achieves a superior result that gives a new perspective for relevant tasks. Extensive experiments are conducted on a large real-world transactional dataset, which shows that our model outperforms other available algorithms (Chapter 5).

1.4 Thesis Structure

The thesis is structured as follow:

Chapter 2 introduces the current methods for the spatio-temporal data impuation and transportation prediction. We first discussed the missing data imputation which is a significant problem when analyzing spatio-temporal data. Second, we group the transportation prediction problem into three categories: time-series models, deep-learning models and latent-space models.

Chapter 3 briefly reviews the related work for the multi-view spatial missing data imputation. A spatially related method is proposed in this chapter, which can only use spatial information to achieve a strong performance. In detail, the method integrates a spatial multi-kernel clustering method and an adaptive-weight non-negative matrix factorization (NMF) for solving the multi-view spatially related tasks. The proposed method is also used to provide complete data for addressing the problem in Chapter 5.

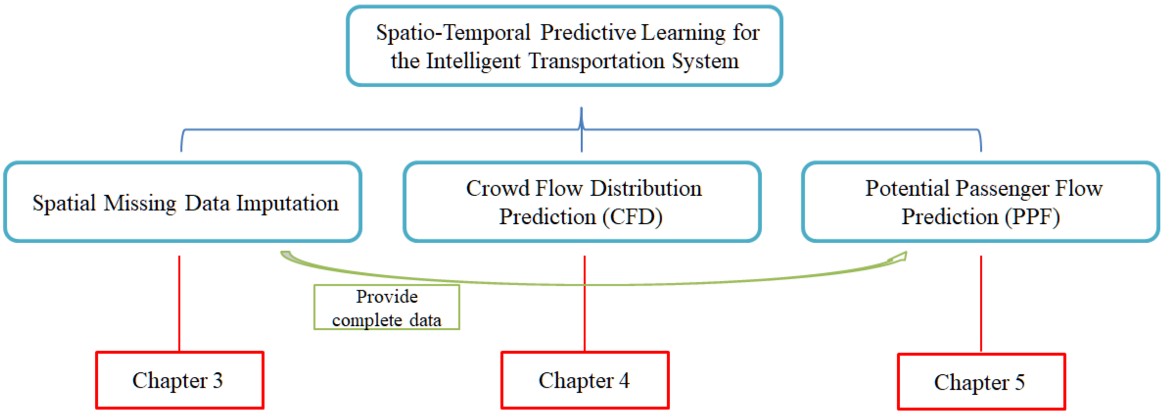

Spatio-Temporal Predictive Learning for the Intelligent Transportation System
Spatial Missing Data Imputation
Crowd Flow Distribution Prediction (CFD)
Potential Passenger Flow Prediction (PPF)
Provide complete data
Chapter 3
Chapter 4
Chapter 5

Figure 1.4: The illustration of the thesis structure.

Chapter 4 briefly reviews current studies on the crowd flow and origin-destination prediction. Then we propose three online latent space (OLS) models. OLS-AO incorporates an average optimization strategy that adapts to stable passenger flows. OLS-MR captures the most recent trends to achieve better performance when sudden changes in crowd flow occur. The dual track model, OLS-DT, integrates both OLS-AO and OLS-MR to exploit the strengths of each model in different scenarios and enhance the models,Äô applicability to real-world situations. Given a series of CFD snapshots, both models learn the latent attributes of the train stations and, therefore, are able to capture transition patterns from one timestamp to the next by combining historic guidance.

Chapter 5 proposes our model on forecasting potential passenger flows. We propose a multi-view localized correlation learning method. The core idea of our strategy is to learn the passenger flow correlations between the target areas and their localized areas with adaptive-weight. To improve the prediction accuracy, other domain knowledge is involved via a multi-view learning process.

The relationship among Chapters 3, 4, and 5 are illustrated in Figure 1.4. Chapter 6 concludes the thesis and outlines the scope of future work.

Chapter 2

Literature Survey

To date, spatio-temporal predictive learning methods have been widely used in the intelligent transportation system, especially in the subway (or city trains) network. We first discuss the missing data completion for both multi-view spatial data and the traffic spatio-temporal data. Then we can group the current predictive learning methods into three categories: time-series models, deep-learning models and latent-space models. In addition, the current studies on the network-wide crowd prediction problem are introduced at the last.

2.1 Missing Data Completion

2.1.1 Spatial Missing Data Imputation

Due to the challenge discussed in Chapter 1.2.1, we first introduce the imputation methods related to spatial data. Missing data imputation is a significant task for data analysis [77]. In the spatially related problem, neighborhood and collaborative filtering [67, 92] based methods are two kinds of dominant approaches in missing data filling. Although some classical methods (e.g., zero-filling, mean value filling, regression models) can be applied to the spatial missing data imputation, they have disadvantages in nature, i.e, they are not designed for this spatial problem. [11] used the inverse distance weighting (IDW) method to interpolate the spatial rainfall distribution. [85] utilized the spatial information as inputs in a residual kriging method to estimate the average monthly temperature. Unlike the spatial model, some successful spatio-temporal models were proposed for use with time stream data [5, 13, 92, 106]. However, they focused on filling missing entries by considering both spatial and temporal properties, and would not perform well on the static spatial data without the temporal information. Furthermore, these discussed methods leveraged the spatial guidance but did not consider the problem on multi-view datasets.

2.1.2 Multi-view Learning

We discuss the multi-view studies because the missing-data usually contain the multiply views. Multi-view learning methods involved the diversity of different views that can jointly optimize functions based on various feature subsets [40, 66]. [88] proposed a matrix co-factorization based method (MVL-IV) to embed different views into a shared subspace, such that the incomplete views can be estimated by the information on observed views. To connect multiple views, MVL-IV assumes that different views have distinct 'feature' matrices (i.e., $\{H_i\}_{i=1}^{d}$), but correspond to the same coefficient matrix (i.e., W). However, it does not exploit the spatial correlations and may suffer from the imbalance problem, i.e., if there is a substantial missing ratio gap between views, the coefficient matrix W is mostly learned from the dense view. The proposed method has addressed this weakness by introducing guidance matrices. Another widely used strategy for solving the multi-view problem is tensor factorization [65, 86], but this restricts a regular tensor that requires the number of dimensions per view to be the same. Moreover, multiple kernel learning with incomplete views [46, 76] only focuses on completing missing kernels instead of filling missing values. To the best of our knowledge, none of the above studies considered both spatial and multi-view problems. Hence, Chapter 3 proposes an effective missing value imputation model for multi-view urban statistical data. There are numerous sensors collecting traffic data in road, but they distribute unevenly and probably change over time. In the extreme case, no data is generated in some roads during a time interval. Besides, malfunctions of sensors and human factors also lead to the problem of missing data. As a result, traffic data can be of high sparsity, which makes missing data completion an important step in traffic prediction.

2.1.3 Missing Data Imputation for Spatio-temporal Data

We also discuss other spatio-temporal traffic missing data imputation methods in this section. Missing data completion aim at filling out the data with estimation value. For a large city network, we are faced with a truth that data is not everywhere, especially in the real-time system, there may be not enough time to collect complete data.

A naive way is to average the values near missing data. Research by [37] used the simple average method to impute the missing data. As they said, this was mainly because the missing ratios for the selected sensors are sufficiently low. However, when faced with large-scale traffic network, the number of missing data is probably huge, and the average method cannot be adopted.

Many traffic prediction methods incorporate missing data completion into prediction steps. [69] dealt with missing data by 'expanded Bayesian network'. They made use of the causal relations in traffic network, and constructed the network by replacing the missing data with its causal variables. The main shortage for this method is that once the structures and parameters of the Bayesian network is trained, the relative position and time for missing data is also fixed. This is usually unreal, because data is likely to be missed at different time and sites. In other words, one model can only handle with one case of missing data. If we are facing a real traffic network, it is impossible for us to enumerate every condition and train for each condition a model.

[15] proposed a data completion method by matrix factorization. The Traffic data were structured as matrix with each entry X_{ij} denotes traffic speed between node i and node j. Based on the non-negative matrix tri-factorization framework, they got the latent attribute matrix of nodes and the attribute interaction matrix. By minimising the known error together with constraint using Laplacian matrix [26], missing data completion was accomplished by reconstructing data matrix with factor matrices.

Tensor decomposition [32] was used by [1] to complete missing computer network traffic data. They used a weighted optimization version of CP decomposition to impute the missing data. [71] improve this method through Tucker decomposition. They got a comparatively accurate result even when the missing ratio of data was quite high (up to 75%). These methods organised data as a three-way tensor, with day mode, hour mode and interval mode. [61] used this method to floating car data and get a better coverage of traffic state. In this paper, data is organised as a three-way tensor of link mode, interval mode and day mode. Similar method was used in research of [72]. They treated data to be predicted as missing data, and trained the decomposition model with historical data as rough prediction. There are two main problems of these tensor-based methods. First, they can only deal with one road or several road segments at a time, which is not enough for a citywide traffic network. Second, they did not define a rule for choosing the ranks for tensor decompositions, but the rank is one of the most crucial parameters for tensor decompositions.

[93] develops a spatio-temporal multi-view approach (ST-MVL) to collectively complete missing values in a collection of geo-sensory time series data. It considers that 1) the temporal correlations between readings at various time spans in the same series and 2) the spatial correlations between different time series.

2.2 Traffic and Crowd Flow Prediction

2.2.1 Time-Series Models

Time-series models treat traffic data as time series. They often build regression models for targeted subway stations. By training with historical data and get optimized parameters, we can get predicted future of passenger flow in targeted stations.

ARIMA is a prevailing parametric model in traffic prediction. It can deal with non-stationary time series such as the traffic flow and entrance/exit crowd flow.

[79] shows a two-layer structure to predict traffic flow of a small road network which consists of four motorways. The first layer uses Kohonen map and spatial information to cluster traffic data into four categories, and the second layer predicts traffic conditions with ARIMA models using same parameters in the same group. This means that if the road network is bigger and more complicated, the number of groups should be much bigger, or the accuracy will drop distinctly. The increment of group number will increase the parameters to be learned, which makes it harder to be implemented in urban areas. [84] also presents a method to predict traffic flow using modified ARIMA model, called seasonal ARIMA. Traffic data often show periodicity, and seasonal ARIMA models make use of this periodicity to improve the accuracy of predictions [52]. Through seasonal differencing, a SARIMA model is built to predict the traffic condition. [55] proposes a parametric and convex optimization-based method, named optimization and prediction with hybrid loss model (OPL). It leverages the linear regression model and the outcomes of seasonal autoregressive integrated moving average (SARIMA) method jointly. It also combined the social media data to improve prediction accuracy under event occurrences. A prediction model combining ARIMA and Kalman filter to predict traffic state is shown by [89]. The state variables of Kalman filter are the historical road traffic volume and speed at the current moment, and the observed measurements are the predicted volume and speed data at the next moment. Another method related to ARIMA is proposed by [91]. After incorporated with a neural network, this method can handle a situation

where data is recorded in a varied interval. This method is more robust compared to other ARIMA based methods.

Other regression model such as SVR is also an effective method used in traffic prediction. A hybrid algorithm is proposed in [70], aiming to predict the short-time crowd flows in Beijing subway network, named Wavelet SVM (SVR). The proposed method take advantages of SVR and Wavelet models. Wavelet transform makes the crowd flow function gradually multi-scale refined, and ultimately achieves high frequency and low frequency information. As we can see, the crowd flow function is finally handled by the wavelet transform with the process of time-frequency signal analysis. [9] proposes to use online-SVR for traffic prediction. [37] introduces a method that used a multi-variable linear regression model as the prediction model. They first built the regression models for the single targeted road whose traffic conditions are to be predicted. Then, they iteratively chose subset of all traffic data on other roads, and trained a multi-variable linear regression model for each subset. Finally, Granger test was performed between the residue of original model and multi-variable models.

It is clear that this method is unsuitable for network-wide traffic prediction, because training models with every subset of traffic data is computationally prohibitive. Like ARIMA models, these methods also built a regression models for traffic data to make predictions. To sum up, the regression model cannot tackle with the crowd flow prediction issue of a large area with a complex road network. They often study a simple subway stations or road network, and train models to fix parameters for each station or road segments, which can be time-consuming especially when a complex network is presented. And the regression models are sensitive to the missing data.

2.2.2 Deep Learning Model

Deep learning models can also be used in traffic and crowd flow prediction tasks. An hourly crowd flow prediction method based on the deep learning models is designed in [42]. The scenario features including inbound and outbound, and smart cards; Temporal features including the hour of a day, the day of a week, and holidays; and the passenger flow features containing the real-time passenger flows and the previous flows, are combined as the input features. These features are designed and trained as different stacked autoencoders (SAE) in the first stage. Then, the pre-trained SAE are further utilized to initialize the supervised DNN with the real-time passenger flow as the label data in the second stage. The hybrid model (SAE-DNN) is applied and evaluated with a case study of passenger flow prediction for four bus rapid transit (BRT) stations of

Xiamen in the third stage. [39] develops a multiscale radial basis function (MSRBF) network for predicting the irregular fluctuation of subway crowd flows.

EMD-BPN is a hybrid method which combines back-propagation neural networks (BPN) and empirical mode decomposition (EMD). It is proposed to forecast the short-term crowd flows in the metro system [83]. There are three steps in the EMD‚ÄìBPN prediction method. The first step (EMD step) decomposes the short-term passenger flow series data into a number of intrinsic mode function (IMF) components. The second step (Component Identification Stage) identifies the meaningful IMFs as inputs for BPN. The third stage (BPN Stage) applies BPN to perform the crowd flow forecasting.

[49] propose a convolutional neural network based model for larger-scale road network speed prediction. This model organises inputs as a matrix to process traffic data as an image. The data in the matrix arranged according to their timestamps and geographical locations. Spatio-temporal correlations of data are then discovered and used by the convolutional neural network.

[96] present a model to predict crowd flows, which is similar to passenger flow prediction. They propose a model named ST-ResNet. In this model, convolutional neural networks and residual networks are used to build a new structure. Through deep residual learning, spatio-temporal correlations of a large area can be used in prediction. Besides, external components such as weather and events are also added in predicting process. Although it performs well in crowd flows prediction, this method may be limited in CFD of subway network.

One problem of deep learning models is that these models are static, while traffic prediction is a dynamic issue. Incidents and events are likely to change traffic patterns of road segments. Changing parameters online for a deep learning network can be difficult and time-consuming, which makes it hard to be implemented in real-time applications. Another problem is the CFD prediction contains many missing entries which leads the above deep learning models cannot be used directly.

2.2.3 Latent-Space Models

Recent years have witnessed an explosive growth of the latent space learning for the applications in network-wide problems, such as in traffic prediction [16], community detection systems [99], and recommendation systems [18]. The benefits of low-rank approximation of latent space model is to remove the redundant information of the large-scale matrix and obtain a more compact matrix [94]. Many studies used latent space strategy in the traffic analysis have achieved the good performance. For example, to solve the problem of discovering and localizing abnormal activities in crowded scenes, Yu et al. designed a spatiotemporal detection model based on Laplacian eigenmap strategy [74].[31] focused on addressing the problem of high-order time-series prediction via a tensor structure. The learned latent core tensor can represent the most important attributes of the original time sequences.

As an effective method of latent space learning, non-negative matrix factorization (NMF) has achieved great performance on solving the network-wide problems. Differ from the original NMF, online latent space (OLS) model is able to capture the temporal transition patterns with time evolves [8, 14]. Recent years, many applicable approaches based on OLS were proposed. Deng et al. [16] proposed an online model that extracts temporal and topological attributes of roads for the network traffic speed prediction. The constrained optimization problem also can be solved by the online latent space learning [6], and Wang et al. [80] took advantage of OLS to handle the large-scale streaming datasets very efficiently. However, due to the particularity of CFD prediction problem, none of above studies can be directly used.

We use the method proposed in [15] as an example. The model can predict traffic speed in a network-scale. They denote a road network as a directed graph, with vertices model road intersections or end of roads, and edges which connect two vertices represent a directed road segment. The prediction model is defined by a node attribute matrix U and a node attribute interaction pattern matrix B, together with a transition matrix A. B can be thought as a matrix that denotes the spatial correlations of road segments. Another matrix A that that approximates the changes of U between timestamps is also introduced. Trained by the method introduced in this article, these matrices can be used to predict traffic conditions of the large-scale road network (up to 8242 vertices and 19986 edges). Moreover, this model is able to impute missing data, and can make predictions on-the-fly. There are two main shortages in this method. Firstly, it only utilises real-time data in a timestamp, without considering any historical data in making predictions. The performance of this model may be improved when considering historical data in making predictions. Secondly, in order to capture the temporal evolution of road attribute matrix U, the model try to learn a transition matrix A from changes of U in last timestamp and current timestamp. Then the model assumes that the changes of U between next timestamp and current timestamp would be the same. These factors restrict the performance of this model.

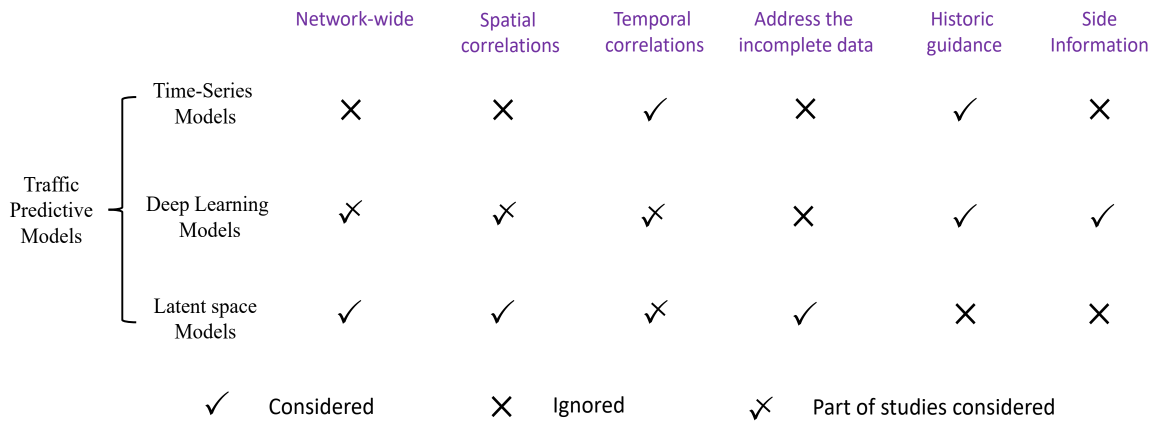
2.2.4 Network-wide Crowd Flow Prediction

In this section, we review the current studies on the network-wide crowd flow and origin-destination prediction.

2.2.4.1 Network-wide Problem

City network-wide crowd flow prediction is not only a significant task for the modern transportation management, but also optimizes many urban services. Nowadays, some of methods were focusing on forecasting the citywide crowd flows which studied the human movement [23, 51, 90, 97]. Ma et al. devised a series of visualization approaches to show the flows' dynamic changes in the networks [51]. Zhang et al. proposed deep learning models based on the ResNet to predict crowd inflows and outflows of the entire city regions [96–98]. The Probabilistic model is an effective approach to estimate the traffic speed. For example, [95] and [41] used trajectory data to estimate citywide traffic volume via probabilistic graphical models. [23] developed a spatial-temporal attention mechanism to predict the city-wide traffic flows. Unfortunately, due to the *real-time delayed data collection* problem, these city traffic prediction methods cannot be utilized to solve our CFD prediction problem directly because they require complete traffic data.

About the metro crowd flow prediction, most existing methods only focused on forecasting entrance/exit flows at certain stations, they neglected the crowd flows across different stations. We list some classical works here [12, 55, 83]. Wei et al. [12] developed an effective short-term passenger flow prediction model to explore the time variants and capture dynamic patterns on a single subway line. Subsequently, a modified approach is proposed based on the neural network, which aims to solve the same entrance/exit crowd flow prediction task in a few metro lines [83]. Ni et al. [55] used auxiliary information, such as social media events, to improve the forecast performance. Sun et al. [70] and Leng et al. [36] took into account a special flow named transfer passenger flow, which describes the volume of passengers transferring from one line to another. The transfer passenger flow are different from CFD, and their studies fail to formulate a network-wide framework. In fact, none of the discussed methods can solve the network-wide problem efficiently because the widely-used time series strategies, such as GPR[107], ARIMA [84] and recurrent neural network [50], are not suitable for the network-wide problem. They usually focus on the flow prediction for one station or few metro lines that will extremely time-consuming if they are applied to the network-wide problem. Other models focused on exploring subway scheduling delay detection and route choice rather than



		Network-wide	Spatial correlations	Temporal correlations	Address the incomplete data	Historic guidance	Side Information
Traffic Predictive Models	Time-Series Models	✗	✗	✓	✗	✓	✗
	Deep Learning Models	✓✗	✓✗	✓✗	✗	✓	✓
	Latent space Models	✓	✓	✓✗	✓	✗	✗

✓ Considered ✗ Ignored ✓✗ Part of studies considered

Figure 2.1: Unified framework for traffic predictive model under current research review.

the network-wide crowd flow forecasting, such as [45, 56, 101]. In conclusion, limited approaches can be used to solve the network-wide CFD prediction directly and efficiently. Besides, none of existing crowd flow prediction methods considered the PPF problem studied in this Thesis. Other relevant studies, such as [28], are point-based prediction model, not in a matrix formulation. [28] selects k points to predict k values. However, in the PPF problem (introduced in Chapter 1.2.3), k target areas require nk prediction values, where n is the number of known areas. It is because we also need to consider the crowd flows between each area.

2.2.4.2 Origin-destination (OD) Prediction

Origin-Destination Prediction aims to estimate the traffic conditions (e.g., travel times, route planning, traffic flows) between origin and destination points, which is a critical task for transportation planning, operation, and management [48]. Roughly speaking, it can be divided into two categories [43]. The first category is the static estimation [2, 25] that focuses on estimating the average OD pairs in a long-term time period, providing advice on transportation development to authorities. The second category is the dynamic model that usually applies to taxi demand prediction and travel time estimation [3, 43, 64, 81, 82, 108]. The most recent research is proposed in [81], which predicts the OD matrix based on a deep learning model via graph convolution strategy. It divides the city area into grids and predicts the passenger demands of given origin grid-areas and destination grid-areas at a given time slot. However, it seems that this method does not consider the *real-time delayed data collection* problem, since it makes all trajectory data observed when testing. A similar problem appears in [64, 108] that they are difficult to solve the real online problem.

In summary, to address our real online CFD prediction problem (introduced in Chapter 1.2.2), we must consider the delayed data collection caused by the ongoing journeys.

2.3 Conclusion

Several methods regarding prediction models are studied, including time-series methods, deep-learning methods and latent-space methods. Time-series models are original method which process traffic data as time series, and methods using time-series models neglect the spatio-temporal correlations in prediction. Methods based on deep learning often consider spatio-temporal correlations of data in an ambiguous way. Besides, deep learning models are often static, while traffic prediction is a dynamic problem. Although shortages exist, deep learning methods generally perform better than time-series models in predictions with large-scale data. Among the methods mentioned in this report, the latent space based method can be regarded as the best method designed for network-wide traffic prediction, in spite of some shortages. This model can predict traffic conditions of a large-scale road network, compared to road networks studied in other articles, and make predictions on-the-fly. It depends on spatial correlations of data to make predictions, and can get at least comparable results with other methods while faster than other methods and much bigger in prediction area. Figure 2.1 shows the unified framework for traffic predictive model under current research review.

CHAPTER 3

Missing Value Imputation for Urban Statistical Data

Large volumes of urban statistical data with multiple views imply rich knowledge about the development degree of cities. These data present crucial statistics which play an irreplaceable role in the regional analysis and urban computing. In reality, however, the statistical data that are divided into fine-grained regions usually suffer from missing data problems. Those missing values hide the useful information that may result in a distorted data analysis. Thus, in this chapter, we propose a spatial missing data imputation method for multi-view urban statistical data. To address this problem, we exploit an improved spatial multi-kernel clustering method to guide the imputation process cooperating with an adaptive-weight non-negative matrix factorization strategy. Intensive experiments are conducted with other state-of-the-art approaches on six real-world urban statistical datasets. The results not only show the superiority of our method against other comparative methods on different datasets, but also represent a strong generalizability of our model.

3.1 Introduction

Urban statistic data connect social sciences, urban computing, administrative management, transportation, and regional planning that are significant for city development [20, 54, 103]. These statistical data usually include multi-fold views (e.g., views of Population and Economy) to reveal the growth gaps among different administrative regions from various perspectives. For example, the economy view records the key economic indicators for fine-grained regions, such as the number of industries and employee statistics; and the population view consists of detailed population information of all age groups in each region.

The statistic data provide key statistics to governments, business and the community on social science, for the benefit of some aspects of human life. However, in some places, statistical data are hard to be entirely acquired due to document defacement, error recordings, and statistician misplay. Such missing data hide useful information which may cause distorted results for further analysis. To the best of our knowledge, it is still a blank field concerning this specific problem, but the real demand appears. Hence, the missing value imputation for urban statistical data is a vital task for reliable urban computing and government services.

In this chapter, we study the problem of missing-data imputation for the Australian Bureau of Statistics (ABS), which has some unique challenges:

• *Missing temporal information.* In the real-world data from ABS, almost all the missing values in the current year were also missing in the past years, which may be caused by the region restriction and complicated human-made errors. This violates the basic assumption of matrix completion [7] that the unobserved entries are sampled uniformly at random. Thus matrix completion-based approaches may not work in this case.

• *Multi-view problem.* The complicated underlying interactions suggest that simply recovering the missing information without considering the correlations among attributes and multi-modes will end up with a poor performance. For example, the economy view has strong correlations with the income and population views, so that a high-quality economy in a region usually goes along with a better income and a larger population; and a low-level economy in a region has a high probability of being connected with a lower income and a smaller population.

• *Spatial correlation mining problem.* As illustrated in Figure 1.1, the statistical data focusing on fine-grained regions may change over locations significantly and non-linearly. Therefore, to properly recover the missing information of statistical data, we need to consider the regional similarities.

To date, a number of missing data imputation approaches are applied in urban statistical data, e.g., mean-filling (MF), k-nearest-neighbor (KNN) filling [57], and collaborative filtering based methods [62]. Most of them, however, have been proposed to focus on the single view problem. Besides, although several spatiotemporal methods can infer the missing information based on the knowledge from both spatial and temporal domains [13, 92, 106], they do not perform well when the missing temporal information challenge appears. To address all challenges, our proposed method is designed as a spatially related method which can only use spatial information to achieve a strong performance. In detail, the method integrates a spatial multi-kernel clustering method and an adaptive-weight non-negative matrix factorization (NMF) for solving the multi-view spatially related tasks. We summarize the main contributions and innovations of this chapter as follows:

- To handle the multi-view problem with spatial characteristic, we propose a Spatially related Multi-Kernel K-Means (S-MKKM) method to identify the underlying relationships among multiple views and capture the regional similarities.

- We propose an adaptive-weight non-negative matrix factorization approach to leverage the information learned above to tackle the multi-view missing data imputation problem. Besides, the proposed method also takes the guidance from the single-view and the real geographic information with KNN strategy into consideration.

- A spatial multi-view missing data imputation method for urban statistical data based on non-negative matrix factorization is proposed, called SMV-NMF. SMV-NMF does not rely on the temporal information but achieves a great performance only using spatial information.

- Our experiments on six real-world datasets verify the effectiveness of our method. All the empirical results show that the proposed method SMV-NMF outperforms all the other state-of-the-art approaches. Furthermore, SMV-NMF shows strong generalizability and can transfer the constructed model from one urban dataset to another well.

3.2 The Proposed Method

Before clarifying our model, we firstly introduce some basic notations, operations and algorithms used in this chapter. The main symbols used in this Chapter are summarized in Table 3.1.

Table 3.1: Symbol description.

Symbols	Descriptions
$X = [X_1 X_2 ... X_d]$	original data matrix contains d views
$W; H_p$	latent space matrices
$Y_p; \bar{Y}_p$	indication matrices for all complete entries and missing entries of p-th view
$k; l$	the number of dimensions of latent space; and the number of clusters
$n; d$	the number of regions; and the number of views
$Z; Z^{'}$	weight matrices
L	graph Laplacian matrix
$X_{mv}; X_{sv}; X_{knn}$	three guidance matrices
K_β	the kernel matrix
$\lambda_1; \lambda_2; \lambda_3; \alpha$	regularization parameters

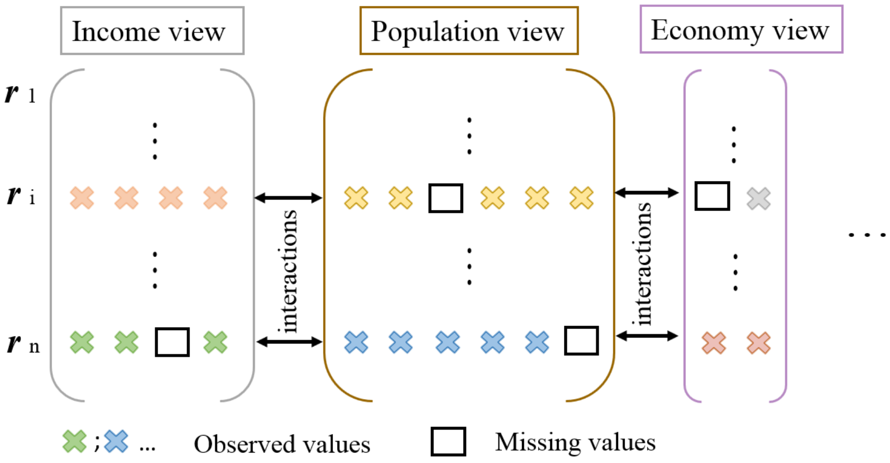


Figure 3.1: Problem description.

3.2.1 Problem Description and Preliminary

As illustrate in Figure 3.1, this research focuses on completing the missing values in the urban statistical data, where one urban dataset contains multiple views, e.g., Income, Population, Economy views, etc. For a dataset with n regions (r_1,...,r_n) and d views, the dimension of attributes in the p-th view is m_p ($1 \leq p \leq d$). Our method aims to impute the missing values with a high accuracy.

3.2.1.1 Multi-view NMF

The multi-view NMF aims to learn a latent subspace $W \in \mathbb{R}_{+}^{n\times k}$ by multiple views $\{X_1...X_d\}$ through the multi-view generation matrices $H_p \in \mathbb{R}_{+}^{k\times m_p}$. The basic missing data imputation model can be described as the following optimization objective:

$$\underset{W\geq 0, H_p \geq 0}{arg\ min}\, J_0 = \sum_{p=1}^{d} ||Y_p \odot (X_p - WH_p)||_F^2, \tag{3.1}$$

where Y_p are indicator matrices whose entry $Y_p(i,j)$ is one if $X_p(i,j)$ has been recorded (for observed values) and zero otherwise (for missing values); and $\odot$ is the Hadamard (element wise) product operator.

3.2.1.2 Multiple Kernel K-means (MKKM)

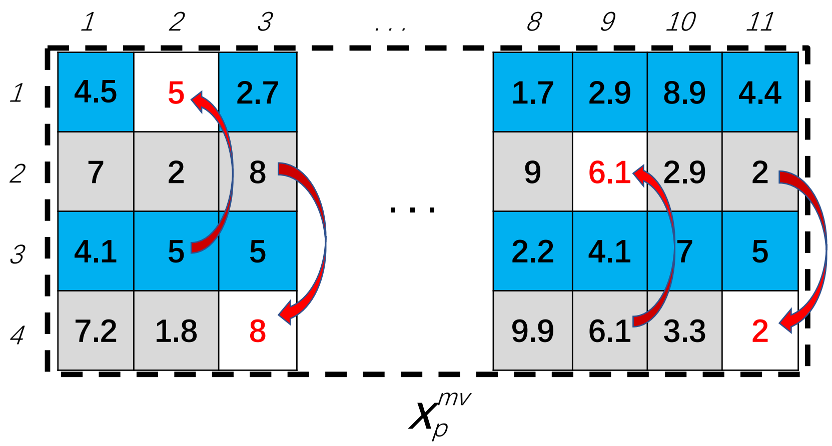
Let $\{\mathbf{x}_i\}_{i=1}^n$ be a collection of n samples (region), $\mathbf{x}_i$ represents the statistical features of the i-th region, and $\phi_p(\cdot)$ be the p-th view mapping that maps $\mathbf{x}$ onto the p-th reproducing kernel Hilbert space. In this case, each sample has multiple feature representations defined by a group of feature mappings $\phi_\beta(\mathbf{x}_i) = [\beta_1\phi_1(\mathbf{x}_i)^\top, \cdots, \beta_d\phi_d(\mathbf{x}_i)^\top]^\top$, where β consists of the coefficients of the d base kernels. A kernel function can be expressed as $\kappa_\beta(\mathbf{x}_i, \mathbf{x}_j) = \phi_\beta(\mathbf{x}_i)^\top\phi_\beta(\mathbf{x}_j) = \sum_{p=1}^d \beta_p^2 \kappa_p(\mathbf{x}_i, \mathbf{x}_j)$. And a kernel matrix K_β is then calculated by applying the kernel function $\kappa_\beta(\cdot,\cdot)$ to $\{\mathbf{x}_i\}_{i=1}^n$. Based on the kernel matrix K_β, the objective of MKKM can be written as:

$$\begin{aligned} &\min_{V,\beta} \mathrm{Tr}(K_\beta(\mathbf{I}_n - VV^\top)) \\ &s.t.\ V \in \mathbb{R}^{n\times l}, V^\top V = \mathbf{I}_l, \beta^\top \mathbf{1}_d = 1, \beta_p \geq 0, \forall p, \end{aligned} \tag{3.2}$$

where V is the clustering matrix; $\mathbf{1}_d \in \mathbb{R}^d$ is a column vector with all 1 elements; $\mathbf{I}_n$ and $\mathbf{I}_l$ are identity matrices with size n and l; l is the number of clusters.

3.2.2 Multi-view Spatial Similarity Guidance

As discussed in Chapter 2.1.2, multi-view matrix factorization based methods suffer from the imbalance problem. In this chapter, we build the similarity guidance X_p^{mv} for the p-th view X_p to address this problem. Accordingly, we propose an approach to obtain regional similarities via the spatially related MKKM model, called S-MKKM. The basic idea is that the development of a city gradually fosters different functional groups, such



	1	2	3	. . .	8	9	10	11
1	4.5	5	2.7		1.7	2.9	8.9	4.4
2	7	2	8	. . .	9	6.1	2.9	2
3	4.1	5	5		2.2	4.1	7	5
4	7.2	1.8	8		9.9	6.1	3.3	2

Figure 3.2: An example of building X_p^{mv}. Assume that regions $\mathbf{x}_1$ and $\mathbf{x}_3$ are falling into one cluster with the blue background, and $\mathbf{x}_2$ and $\mathbf{x}_4$ belong to another cluster with gray background. $\mathbf{x}_2$ and $\mathbf{x}_3$ are the centroid regions of two clusters, respectively. For a missing entry x_{12}, its corresponding value x_{32} is used as an imputation guide. Moreover, if the value in centroid region is missed, then a greedy strategy is implemented to find the nearest observed value (use x_{49} to fill x_{29}).

as educational and business districts, where the regions belonging to the same group would have strong connections with each other [103]. S-MKKM utilizes the MKKM clustering algorithm combined with a graph Laplacian dynamics strategy (an effective smoothing approach for finding spatial structure similarity [15, 22]) to cluster regions into the functional groups. Specifically, we construct a graph Laplacian matrix L, defined as $L = D - M$, where M is a graph proximity matrix that is constructed from the regional physical topology (i.e., $M_{(i,j)} = 1$ if and only if the region $\mathbf{x}_i$ is contiguous to $\mathbf{x}_j$), and D is a diagonal matrix $D_{(i,i)} = \sum_j (M_{(i,j)})$. With this constraint, the S-MKKM model is expressed as follows:

$$\begin{aligned} &\min_{V,\beta} \mathrm{Tr}(K_\beta(\mathbf{I}_n - VV^\top)) + \alpha \mathrm{Tr}(V^\top L V) \\ &s.t.\ V \in \mathbb{R}^{n \times l}, V^\top V = \mathbf{I}_l, \beta^\top \mathbf{1}_d = 1, \beta_p \geq 0, \forall p, \end{aligned} \tag{3.3}$$

where α is the regularization parameter; V is the consensus clustering matrix.

To get the complete kernels, we initially impute the missing data for each view by a simple method, such as KNN or MF. After that, Equation (3.3) can be solved by alternately updating V and β: **i)** With the kernel coefficients β fixed, V can be obtained by choosing the l smallest eigenvectors of $(-K_\beta + \alpha L)$. **ii)** With V fixed, β can be optimized via solving the quadratic programming with linear constraints [46].

The objective of the S-MKKM is to discover the regions with similar properties and build the guidance matrices X_p^{mv}. After having gotten V, X_p^{mv} can be built. Figure 3.2 shows an example of this process. The construction process of X_p^{mv} is that **i)** for the unknown entry x_{ij}, and the region $\mathbf{x}_i \in c$-th cluster, we use its corresponding value $x_{c(i),j}$ from the centroid region to impute x_{ij}; **ii)** if the corresponding value of centroid region is also missed, a greedy strategy will be used to find the nearest observed value for imputation.

3.2.3 Adaptive-Weight NMF

To learn the knowledge from X_p^{mv} more reliably, we propose an adaptive weighting strategy in the NMF imputation process. The adaptive-weight matrix of the p-th view is denoted as $Z_p \in \mathbb{R}_+^{n\times m_p}$, which is built by an exponential function as shown in Equation (3.4) and (3.5).

$$z_{p(i)} = e^{-Dist(\mathbf{v}_i,\mathbf{v}_{c(i)})}, \tag{3.4}$$

$$Z_p = z_p 1_{m_p}^{\top}, \tag{3.5}$$

where $Dist(\cdot,\cdot)$ is the Euclidean distance calculating from the geo-location ($\mathbf{v}_i$) and its corresponding centroid region ($\mathbf{v}_{c(i)}$), here we use the latent embedding $\mathbf{v}_i$ to represent the geo-location of region i, and $\mathbf{v}_{c(i)}$ represents the centroid of the c-th cluster which contains region $\mathbf{v}_i$; $z_p \in \mathbb{R}_+^n$ is a column vector and 1_{m_p} is all-ones vector with size m_p. It is not a straight way for imputation, but the adaptive-weight matrix Z_p controls how much information can be extracted. Z_p adjusts the penalty of each estimated entry. As emphasised in the First Law of Geography [75], the near things have more spatial correlations than distant things. If the distance between $\mathbf{x}_i$ and $\mathbf{x}_{c(i)}$ is small, we want a high penalty to guide the imputation process.

Combining the above strategy, our model can be described as the following optimization function:

$$\underset{W\geq 0, H_p\geq 0}{arg\ min}\, J_1 = J_0 + \lambda_1 \sum_{p=1}^{d} ||\bar{Y}_p \odot Z_p \odot (X_p^{mv} - WH_p)||_F^2, \tag{3.6}$$

where $\bar{Y}_p = \mathbf{1} - Y_p$, $\mathbf{1}$ is an all one matrix that has the same size as Y_p; X_p^{mv} is a homomorphic matrix of X_p; and λ_1 is the regularization parameter to control the learning rate of X_p^{mv}.

3.2.4 Improved by Single-view and KNN Guidances

S-MKKM aims to find the regional groups by considering multiple views simultaneously. However, it is obvious that each view has its characteristics, and the relationships between regions in one specific view are also critical for imputing missing entries. To consider the above knowledge, we apply the spatially related kernel k-means (S-KKM) to capture the similarities among regions of each view. It is essentially analogous to the learning process of S-MKKM as discussed in Chapter 3.2.2, but considering each view, respectively. For one view X_p, the S-KKM model is expressed as follows:

$$\begin{aligned} &\min_{V_p} \ \mathrm{Tr}(K_p(\mathbf{I}_n - V_p {V_p}^\top)) + \alpha \mathrm{Tr}(V_p^\top L V_p) \\ &\quad s.t. \ V_p \in \mathbb{R}^{n\times l}, {V_p}^\top V_p = \mathbf{I}_l, \end{aligned} \tag{3.7}$$

where K_p is one separate kernel and V_p represents the p-th clustering matrix based on X_p.

In fact, to reduce the complexity of our model, we assume that the physical location affects the clustering performance with the same degree and the number of clusters is the same as that in S-MKKM, i.e., l and α are the same as used in Equation (3.3). The reason behind this assumption is that most cities have the same functional regions, such as the residential region and business region. Thus, it is reasonable that we choose the same α and l in this practical task. Besides, α and l are very stable due to the intrinsic property of the urban statistical data, and we fixed them in the experiments. The single view guidance matrix X_p^{sv} and adaptive-weight matrix $Z_p^{'}$ can be constructed by the same strategy of building X_p^{mv} and Z_p.

Furthermore, for each region, its k-nearest spatial neighbors imply rich information that should be considered in our model. Even though the regional physical topology is already involved in multi-view and single-view learning processes, the KNN is a more flexible method. After structuring X_p^{knn} which is an imputed matrix with the average value of k-nearest neighbors, our final optimization function is shown as follows:

$$\begin{aligned} \arg\min_{W\geq 0, H_p \geq 0} \ J = J_1 + \lambda_2 \sum_{p=1}^{d} ||\bar{Y_p} \odot Z_p^{'} \odot (X_p^{sv} - WH_p)||_F^2 \\ + \lambda_3 \sum_{p=1}^{d} ||\bar{Y_p} \odot (X_p^{knn} - WH_p)||_F^2, \end{aligned} \tag{3.8}$$

where λ_2 and λ_3 are the regularization parameters to control the learning rate of X_p^{sv} and X_p^{knn}, respectively.

Given the estimated factor matrices W and H_p based on the above update equations, the filled data are given by:

$$\hat{X_p} = Y_p \odot X_p + \bar{Y_p} \odot (WH_p). \tag{3.9}$$

3.2.5 Learning Algorithm

As Equation (3.8) is a non-convex problem, we use the multiplicative update strategy [35] to ensure the convergence under the following update rules. We first initialize latent space matrices (W and H_p) by decomposing data matrices $\{X_1 ... X_d\}$. The update rules for W and H_p are presented in Equation (3.10) - (3.11).

$$W = W \odot \frac{\sum_{p=1}^{d} (Y_p \odot X_p + \bar{Y_p} \odot (\lambda_1 Z_p \odot \ X_p^{mv} + \lambda_2 Z_p^{'} \odot X_p^{sv} + \lambda_3 X_p^{knn}))H_p^{\top}}{\sum_{p=1}^{d} ((Y_p + \bar{Y_p} \odot (\lambda_1 Z_p + \lambda_2 Z_p^{'} + \lambda_3 \mathbf{1})) \odot (W^{\top} H_p) H_p^{\top})}, \tag{3.10}$$

$$H_p = H_p \frac{W(Y_p \odot X_p + \bar{Y_p} \odot (\lambda_1 Z_p \odot X_p^{mv} + \lambda_2 Z_p^{'} \odot X_p^{sv} + \lambda_3 X_p^{knn}))}{W(Y_p + \bar{Y_p} \odot (\lambda_1 Z_p + \lambda_2 Z_p^{'} + \lambda_3 \mathbf{1})) \odot (W^{\top} H_p)}. \tag{3.11}$$

The above two multiplicative update rules guarantee to be non-negative if the initialization is positive. Without this constraint, the matrices W and H_p could be negative, thus the imputation results could be negative too, which is a contradiction to the facts. We now derive the update rule of W as an example, other variables can be solved with a similar process. The objective of J could be rewritten as follows:

$J = L_0 + L_1 + L_2 + L_3$, where:

$$\begin{aligned}
L_0 &= \sum_{p=1}^{d} ||Y_p \odot (X_p - WH_p)||_F^2, \\
L_1 &= \lambda_1 \sum_{p=1}^{d} ||\bar{Y_p} \odot Z_p \odot (X_p^{mv} - WH_p)||_F^2, \\
L_2 &= \lambda_2 \sum_{p=1}^{d} ||\bar{Y_p} \odot Z_p^{'} \odot (X_p^{sv} - WH_p)||_F^2, \\
L_3 &= \lambda_3 \sum_{p=1}^{d} ||\bar{Y_p} \odot (X_p^{knn} - WH_p)||_F^2.
\end{aligned} \tag{3.12}$$

We provide the derivative of L_0 respect to W as an example, the other components can be derived in the same way. L_0 could also be rewritten as follows:

$$L_0 = \langle Y_p \odot (X_p - WH_p), Y_p \odot (X_p - WH_p) \rangle, \tag{3.13}$$

where $\langle , \rangle$ presents the inner product of matrix. Then:

$$\begin{aligned} dL_0(W) &= -2 \sum_{p=1}^{d} \langle dWH_p, Y_p \odot (X_p - WH_p) \rangle \\ &= -2 \sum_{p=1}^{d} \langle dW, Y_p \odot (X_p - WH_p) H_p^\top \rangle \\ \Rightarrow \frac{\partial L_0}{\partial W} &= -2 \sum_{p=1}^{d} Y_p \odot (X_p - WH_p) H_p^\top . \end{aligned} \tag{3.14}$$

Analogously, we can get:

$$\frac{\partial L_1}{\partial W} = -2\lambda_1 \sum_{p=1}^{d} \bar{Y_p} \odot Z_p \odot (X_p^{mv} - WH_p) H_p^\top , \tag{3.15}$$

$$\frac{\partial L_2}{\partial W} = -2\lambda_2 \sum_{p=1}^{d} \bar{Y_p} \odot Z_p^{'} \odot (X_p^{sv} - WH_p) H_p^\top , \tag{3.16}$$

$$\frac{\partial L_3}{\partial W} = -2\lambda_3 \sum_{p=1}^{d} \bar{Y_p} \odot (X_p^{knn} - WH_p) H_p^\top . \tag{3.17}$$

To utilize the multiplicative update strategy [35], we set the step γ to:

$$\gamma = \frac{W}{\sum_{p=1}^{d} ((Y_p + \bar{Y_p} \odot (\lambda_1 Z_p + \lambda_2 Z_p^{'} + \lambda_3 \mathbf{1})) \odot (W^\top H_p) H_p^\top)}, \tag{3.18}$$

then, we got the update rule of W as shown in Equation 3.10. Besides, the update rule of H_p can be obtained via a similar derivation process.

The process of SMV-NMF is summarized in Algorithm 1.

```
Algorithm 1: SMV-NMF
Input: original data {$X_p$ }; graph Laplacian matrix $L$.
Output: complete data {$\hat{X_p}$}.
Impute $X_p$ by KNN for an initialization.
Initialize $W$ and $H_p$ by decomposing $X_p$.
Construct $X_p^{mv}$, $X_p^{sv}$ and $X_p^{knn}$ by S-MKKM, S-KKM, and KNN respectively.
for $t$ = $1$ to $T$ do
    if $|J_t - J_{t+1}|$ / $J_t \geq \varepsilon$ then
        update $W$ By Equation (3.10)
        update $H$ By Equation (3.11)
    else
        Break
Return $\hat{X_p}$ By Equation (3.9).
```

3.2.6 Time complexity and convergence

We discuss the time complexity and convergence of SMV-NMF here. The time complexity of guidance matrices X_p^{mv} and X_p^{sv} is mainly affected by MKKM. Even though MKKM has a high computational complexity ($O(n^3)$), it is not involved in update loop of variables (W and H_p). Equation (3.10) and Equation (3.11) present that the time complexity of our final function is governed by matrix multiplication operations in each iteration. Therefore, the time complexity per iteration is dominated by $O(nk^2)$. Due to the pursuing of pinpoint accuracy, we sacrifice efficiency to some degree in this real-world problem. In terms of convergence, Algorithm 1 is guaranteed to converge when W or H_p is fixed, because the second-order derivatives regarding W or H_p are positive semi-definite. Thus, the objective function can achieve its optimal value by optimizing W and H_p alternately.

3.2.6.1 Proof of Convergence

First part of convergence. The aim of this part is to find anauxiliary function for SMV-NMF objective function as expressed in Equation (3.8).

Definition 1. *$G(h,h^{'})$ is an auxiliary function for our final function $J(h)$ if the following conditions are satisfied:*

$$G\left(h',h\right) \geq J(h) \quad \text{and} \quad G(h,h) = J(h). \tag{3.19}$$

The auxiliary function is useful because of the following lemma, and the proof of Lemma 3.1 is given by [35].

Lemma 3.1. *If G is an auxiliary function, then J is non-increasing under the update:*

$$h^{t+1} = \arg\min_{h} G\left(h, h^t\right), \tag{3.20}$$

consequently, we have:

$$J\left(h^{t+1}\right) \leq G\left(h^{t+1}, h^t\right) \leq G\left(h^t, h^t\right) = J\left(h^t\right). \tag{3.21}$$

Lemma 3.2. *If $K(h^t)$ is a diagonal matrix under the following definition,*

$$K(h^t) = diag(W diag(v) W^T h./h), \tag{3.22}$$

where v is a column vector of $V = Y_p + \overline{Y}_p \odot \left(\lambda_1 Z_p + \lambda_2 Z'_p + \lambda_3 \mathbf{1}\right)$ then,

$$\begin{aligned} G\left(h, h^t\right) =& J\left(h^t\right) + \left(h - h^t\right)^T \nabla J\left(h^t\right) \\ &+ \frac{1}{2}\left(h - h^t\right)^T K\left(h^t\right)\left(h - h^t\right), \end{aligned} \tag{3.23}$$

is an auxiliary function for $J(h)$.

Proof. Since $G(h,h) = J(h)$ is obvious, we need only show that $G(h,h^t) \geq J(h)$. To do this, we compare

$$\begin{aligned} J(h) =& J\left(h^t\right) + \left(h - h^t\right)^T \nabla J\left(h^t\right) \\ &+ \frac{1}{2}\left(h - h^t\right)^T \left(W diag(v) W^T\right)\left(h - h^t\right), \end{aligned} \tag{3.24}$$

with Equation (3.23) to find that $G(h,h^t) \geq J(h)$ is equivalent to

$$0 \leq \left(h - h^t\right)^T \left[K\left(h^t\right) - W diag(v) W^T\right]\left(h - h^t\right). \tag{3.25}$$

The next step is to prove $\left[K\left(h^t\right) - W\,\mathrm{diag}(v) W^T\right]$ is positive semi-definite. Let $Q = W diag(v) W^T$, then $\left[K\left(h^t\right) - W\,\mathrm{diag}(v) W^T\right]$ can be expressed as $[diag(Qh./h) - Q]$. As the Lemma 3.1 provided in [24], if Q is a symmetric non-negative matrix and h be a positive vector, then the matrix $\hat{Q} = diag(Qh./h) - Q \geq 0$.

■

Second part of convergence. We can now demonstrate the convergence of method.

Proof. Replacing $G(h,h^t)$ in Equation (3.20) by Equation (3.23) results in the update rule:

$$h^{t+1} = h^t - K\left(h^t\right)^{-1} \nabla J\left(h^t\right). \tag{3.26}$$

Since Equation (3.23) is an auxiliary function, J is nonincreasing under this update rule, according to Lemma 3.1. Writing the components of this equation explicitly, we obtain

$$h_a^{t+1} = h_a^t \frac{(Wx)_a}{\left(W(v \odot W^T h)\right)_a}, \tag{3.27}$$

where x is the column vector of $X = \left(Y \odot X + \overline{Y} \odot \left(\lambda_1 Z \odot X^{mv} + \lambda_2 Z' \odot X^{sv} + \lambda_3 X^{knn}\right)\right)$.

By reversing the roles of W and H in Lemma 3.1 and 3.2, J can similarly be shown to be nonincreasing under the update rules for W.

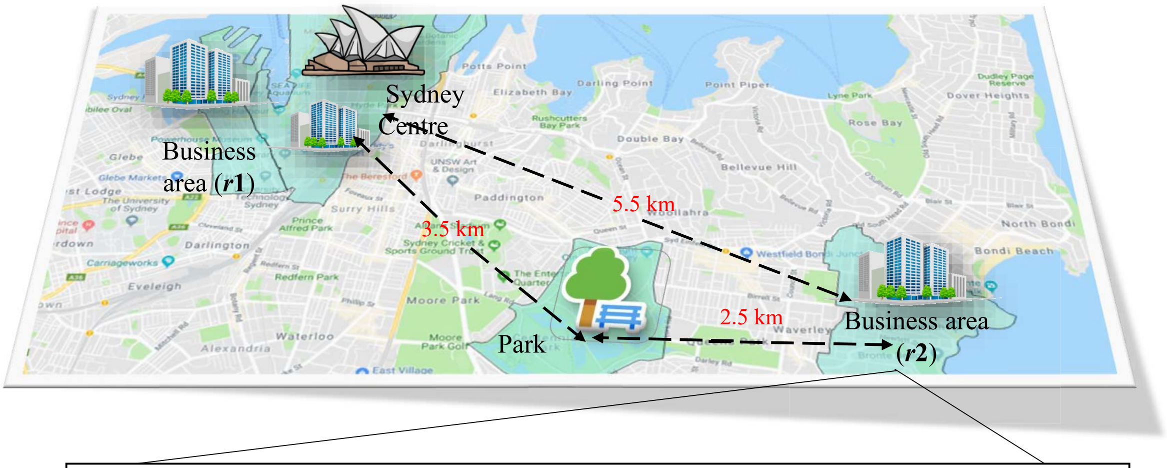
■

3.3 Experiments

In this chapter, we have conducted comprehensive experiments to demonstrate the effectiveness of our method.

3.3.1 Datasets

There are six real-world urban statistical datasets (**Sydney**, **Melbourne**, **Brisbane**, **Perth**, **SYD-large**, and **MEL-large**), where **-large** datasets contain much more fine-grained regions from Australian Bureau of Statistics (2017). Each dataset contains four views, i.e., Economy, Family, Income, and Population. The data example is shown in the Figure 3.3. As we can see in the example, the data of each area contains four views. For example, the economy view can reflect the current business status via total number of businesses, value of total building, etc. The size of the six datasets are 174, 284, 220, 130, 2230, 1985 respectively. The designation of regions is based on the Australian Statistical Geography Standard for the best practical value. The scales of different views are normalized into the same range [0,10] so that we can evaluate the results together.



Economy View	Total number of businesses	Total number of businesses entries	Total number of businesses exits	Value of total building	...
	11,843	2,027	1,496	615	...
Family View	Total households	Total families	Average Family Size	Separate house	...
	7417	6074	3	3203	...
Income View	Mean Employee income	Mean Investment income	Mean Superannuation and annuity income	Median Employee income	...
	56923	7808	26200	53489	
Population View	Person Total	Working Age Population %	Persons/km2	Australian citizen %	...
	132,733	67.8	3909.1	81.8	...

Figure 3.3: The example of ABS data and visualization.

The numbers of the dimension of the four views are 43, 44, 50, 97, respectively. We choose Australian cities mostly because the Australian Bureau of Statistics provides enough data for our study, while such data from other countries is inaccessible to us. However, our method is general enough and can be applied to other cities with administrative areas and statistical census data. To guarantee the diversity of testing, for each missing ratio, we randomly select the test columns and repeat the experiment 20 times and report average results.

3.3.2 Baselines & Measures

3.3.2.1 Baselines

We compare the proposed method SMV-NMF with the following 12 baselines. All parameters of the proposed method and baselines are optimized by the grid search method.

sKNN: A classical method that uses the average values of its k nearest spatial neighbors as an estimate (k=6).

MKKMIK$^{\mathbf{a}}$: A MKKM based method to handle the incomplete views [46]. We modified it to adapt to the spatially related data, then interpolated a missing value by its k nearest spatial neighbors (k=6);

MKKMIK$^{\mathbf{b}}$: Similar to MKKMIK$^{\mathbf{a}}$ but utilize the mean value of each cluster to fill the missing data.

NMF: Fill the missing data by NMF.

IDW: A global spatial learning method compared in many works [11, 13].

UCF: The Local spatial learning method based on collaborative filtering [67, 92].

IDW+UCF: The average result of IDW and UCF.

MVL-IV: A state-of-the-art multi-view learning method based on matrix co-factorization, which learns a same coefficient matrix to connect multiple views [88].

ST-MVL: A state-of-the-art method to impute spatio-temporal missing data [92]. We only use its spatial part due to the problem of missing temporal information.

SMV-MF; MV-NMF$^{\mathbf{a}}$; MV-NMF$^{\mathbf{b}}$: Remove the non-negativity constraint in SMV-NMF; Remove the graph Laplacian dynamics strategy in SMV-NMF when building the X_p^{mv} and X_p^{sv}; Remove the KNN guidance in SMV-NMF.

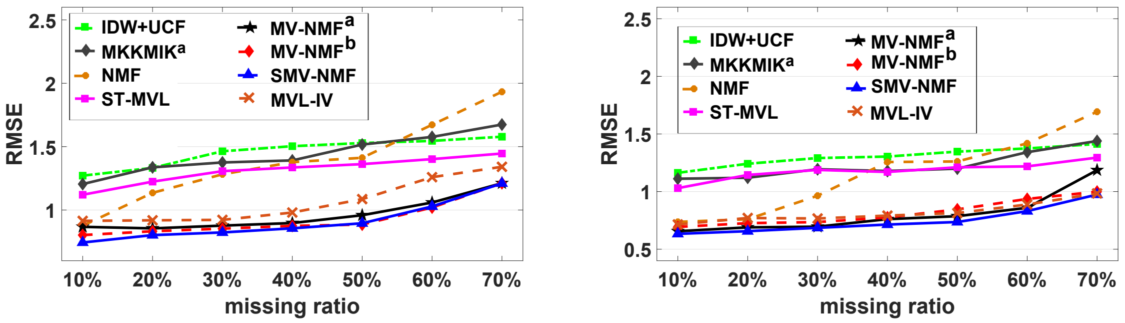
Measures. We utilized the most widely used evaluation metrics in this chapter, namely Mean Relative Error (MRE) and Root Mean Square Error (RMSE).

$$MRE = \frac{\sum_{i=1}^{Q} |u_i - \hat{u}_i|}{\sum_{i=1}^{Q} u_i}, \quad RMSE = \sqrt{\frac{\sum_{i=1}^{Q} (u_i - \hat{u}_i)^2}{Q}},$$

where $\hat{u}_i$ is a prediction for missing value, and u_i is the ground truth; Q is the number of prediction values.

Methods	Sydney		Melbourne		Brisbane		Perth		SYD-large		MEL-large	
	MRE	**RMSE**	**MRE**	**RMSE**	**MRE**	**RMSE**	**MRE**	**RMSE**	**MRE**	**RMSE**	**MRE**	**RMSE**
sKNN	0.3302	1.5319	0.3108	1.3181	0.3534	1.4787	0.3701	1.5754	0.2998	1.2543	0.2635	1.1155
MKKMIK^b	0.3281	1.5507	0.3462	1.4635	0.3773	1.5934	0.3986	1.6992	0.3413	1.6112	0.3067	1.4552
IDW	0.3321	1.5183	0.3187	1.3188	0.3517	1.4663	0.3724	1.5574	0.3273	1.4992	0.3081	1.2587
UCF	0.3566	1.6631	0.3380	1.4635	0.3626	1.5928	0.3757	1.6554	0.3327	1.4230	0.3321	1.5093
IDW+UCF	0.3300	1.4604	0.3141	1.3048	0.3408	1.3967	0.3591	1.4924	0.3045	1.2236	0.2970	1.2111
MKKMIK^a	0.3073	1.4393	0.2833	1.2264	0.3167	1.3479	0.3546	1.5066	0.2915	1.2808	0.3019	1.2300
NMF	0.2189	1.3841	0.1990	1.1557	0.2225	1.3048	0.2469	1.2866	0.2385	1.1996	0.2032	1.0660
ST-MVL	0.2948	1.3137	0.2833	1.1796	0.3117	1.2932	0.3325	1.3949	0.2948	1.0772	0.2829	1.1453
MVL-IV	0.1948	1.0603	0.1744	0.8185	0.1970	0.9698	0.2252	1.0676	0.1792	0.8959	0.1834	0.9223
SMV-MF	0.1911	0.9360	0.1851	0.8006	0.1832	0.8033	0.2199	0.9647	0.1777	0.8315	0.1922	0.9015
MV-NMF^a	0.1806	0.9257	0.1816	0.8159	0.1640	0.7296	0.2170	0.9721	0.1714	0.8226	0.1858	0.8613
MV-NMF^b	0.1829	0.9609	0.1738	0.8048	0.1647	0.7703	0.2239	1.0095	0.1681	0.8046	0.1763	0.8124
SMV-NMF	**0.1773**	**0.9084**	**0.1687**	**0.7471**	**0.1574**	**0.7051**	**0.2097**	**0.9347**	**0.1620**	**0.7753**	**0.1692**	**0.7911**

Table 3.2: The average MRE and RMSE of all missing ratios on four urban statistical datasets. Best results are bold.



(a) Test on Sydney.

(b) Test on Melbourne.

Figure 3.4: Average RMSE with the variation of missing ratios.

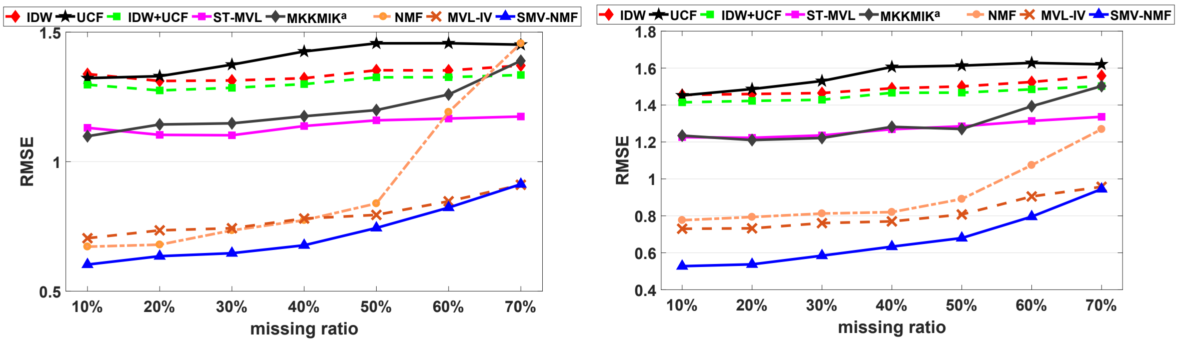
3.3.3 Results on Urban Statistical Datasets

The first set of experiments is designed to assess performance on each dataset. We pick up half of statistical fields (properties) in each urban dataset randomly as the validation set, and the other half as the test set. In the test set, we randomly select missing ratios from 10% to 70% to evaluate the imputation accuracy.

Table 3.2 presents the average errors of all missing ratios across different test methods. It is clear show that our approaches (SMV-MF, MV-NMF^a, MV-NMF^b, SMV-NMF) perform much better than other baselines across different missing ratios on six real-world datasets, where SMV-NMF achieves the best results. Without the non-negativity constraint, SMV-MF performs worse than SMV-NMF, which demonstrates the effectiveness of this constraint. MVL-IV yields better results than ST-MVL, MKKMIK$^{\mathbf{a}}$, IDW+UCF, and NMF becuase it considers the multi-view problem.

To represent our results more clearly, we pick the top eight methods varying different missing ratios on the Sydney and Melbourne datasets, which is shown in Figure 3.4. It is apparent that NMF is sensitive to the missing ratio, which could get good results under the lower level missing ratios, but performs worse when the missing ratio increases. Our methods, (SMV-MF, MV-NMF^a, MV-NMF^b, SMV-NMF) have significant improvements compared with current baselines.

Overall, SMV-NMF outperforms the other baselines because it integrates both multi-view and spatial problems to address the specified missing data imputation task. MV-NMF^a and MV-NMF^b remove a part of the spatial guidance which results in slightly worse performances than SMV-NMF.



(a) Test dataset Melbourne.

(b) Test dataset Brisbane.

Figure 3.5: The average RMSE in generalizability tests.

3.3.4 Generalizability Test

We conduct experiments on testing the generalizability in this section. In detail, we choose the dataset Sydney as the validation set and two urban datasets (Melbourne and Brisbane) as the test sets. We report the experimental results on eight available algorithms. SMV-NMF is the most outstanding approach, as shown in Figure 3.5.

Our method represents strong generalizability which can transfer the constructed model from one urban dataset to another. This is because there are high correlations among cities. For example, the number of functional regions of each city is mostly the same, resulting in the same amount of clusters. The gap between SMV-NMF and MVL-IV narrows as the missing ratio increases, but the former is more robust than the latter because SMV-NMF achieves the best results across all missing ratios. Table 3.3 reveals the average errors using two evaluation metrics. The generality test demonstrates that our model SMV-NMF is a universal model that performs well crossing different urban statistical datasets.

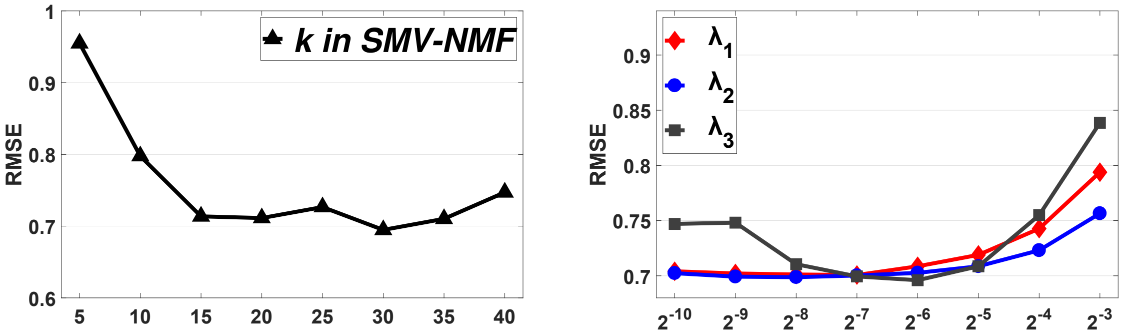
3.3.5 The Sensitivity of Parameters

This section evaluates the performances of SMV-NMF by varying the critical parameters (k, λ_1, λ_2, and λ_3). Due to space limitations, we have only shown the experimental results for the Sydney validation dataset. We discuss them separately but pick them up by the grid search method because four parameters have high dimensional correlations that are hard to visualize. Our illustration approach that discusses parameters separately has been widely used in many other research papers [15, 22].

Figure 3.6 (a) shows the different performances with a varying setting for k. When

Methods	Dataset Melbourne		Dataset Brisbane	
	MRE	**RMSE**	**MRE**	**RMSE**
UCF	0.3311	1.4026	0.3656	1.5624
IDW	0.3324	1.3374	0.3697	1.4934
IDW+UCF	0.3182	1.3061	0.3518	1.4552
MKKMIK[a]	0.2827	1.2018	0.3137	1.3020
ST-MVL	0.2794	1.1391	0.3123	1.2698
NMF	0.1538	0.9067	0.1781	0.9196
MVL-IV	0.1510	0.7879	0.1636	0.8089
SMV-NMF	**0.1506**	**0.7202**	**0.1493**	**0.6718**

Table 3.3: Generalizability test. We report the average MRE and RMSE of all missing ratios and best results are bold.



(a) Factor k.

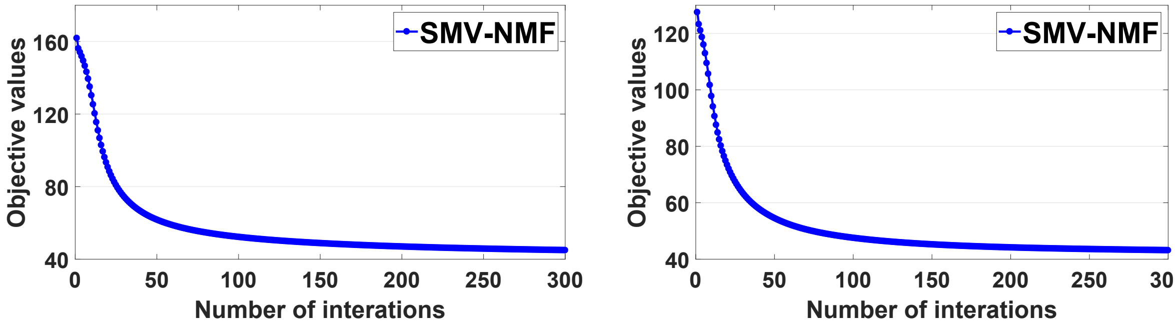
(b) Factor λ_1&λ_2&λ_3.

Figure 3.6: Effect of Parameters.

we increase k from 5 to 15, the results improve significantly. However, the performance tends to stay stable at $15 \leq k \leq 35$. In particular, SMV-NMF achieves the best result when $k = 30$, while it can get good performance if the k is set between 15 and 35. This indicates that a low-rank latent space representation can already capture the attributes of the urban statistical data.

Figure 3.6 (b) reveals the effect of varying λ_1, λ_2, and λ_3. These three parameters determine the strength of the three guidance matrices X_{mv}, X_{sv}, and X_{knn}, respectively. λ_1=2^{-7}, λ_2=2^{-8} and λ_3=2^{-6} yield the best results for SMV-NMF. We observe that the performance is stable when these three parameters are ranged between 2^{-8} and 2^{-6}.

In summary, both parameters used in this chapter bring benefits to the improvement of our models. Furthermore, our model is stable and easy fine-tuning because it is insensitive to these parameters.



(a) Conduct on the largest dataset (Melbourne).

(b) Conduct on the smallest dataset (Perth).

Figure 3.7: Convergence rate.

Table 3.4: Effects of different initialization methods.

	Zero-init	Random-init	Mean-init	KNN-init
RMSE	1.1741	1.1347	0.9496	0.9084

3.3.6 Initialization and Convergence

To get the complete kernels, we first impute the missing data for each view by an efficient method, such as KNN and MF. The effects of different initializations are reported in Table 3.4. Based on the results, we easily find that the initialization method KNN could achieve the great performance for SMV-NMF. Accordingly, we choose the KNN method for a good balance between time-consuming and accuracy.

Figures 3.7 (a) and (b) show the convergence trends of iterative model SMV-NMF on both the largest and smallest datasets. It illustrates that our algorithm can converge into a local solution in terms of the objective value in a small number of iterations.

3.4 Conclusion

In this chapter, we propose a spatial missing data imputation method for multi-view urban statistical data, called SMV-NMF. To address the multi-view problem, an improved spatial multi-kernel method is designed to guide the imputation process based on the NMF strategy. Moreover, the spatial correlations among different regions are involved in our method from two perspectives. Firstly, the latent similarities are discovered by S-MKKN and S-KKM based on the idea of finding functional regions, and secondly, KNN is used for capturing the information of real geographical positions. We conduct intensive experiments on six real-world datasets to compare the performance of our model and other state-of-the-art approaches. The results not only show that our approach outperforms all other methods, but also represent strong generalizabilities crossing different urban datasets.

CHAPTER 4

Crowd Flow Distribution Prediction

As a key mission of the modern traffic management, crowd flow prediction (CFP) benefits in many tasks of intelligent transportation services. However, most existing techniques focus solely on forecasting entrance and exit flows of metro stations that do not provide enough useful knowledge for traffic management. In practical applications, managers desperately want to solve the problem of getting the potential passenger distributions to help authorities improve transport services, termed as crowd flow distribution (CFD) forecasts. Therefore, to improve the quality of transportation services, we proposed three spatiotemporal models to effectively address the network-wide CFD prediction problem based on the online latent space (OLS) strategy. Our models take into account the various trending patterns and climate influences, as well as the inherent similarities among different stations that are able to predict both CFD and entrance and exit flows precisely. In our online systems, a sequence of CFD snapshots is used as the training data. The latent attribute evolutions of different metro stations can be learned from the previous trend and do the next prediction based on the transition patterns. All the empirical results demonstrate that the three developed models outperform all the other state-of-the-art approaches on three large-scale real-world datasets.

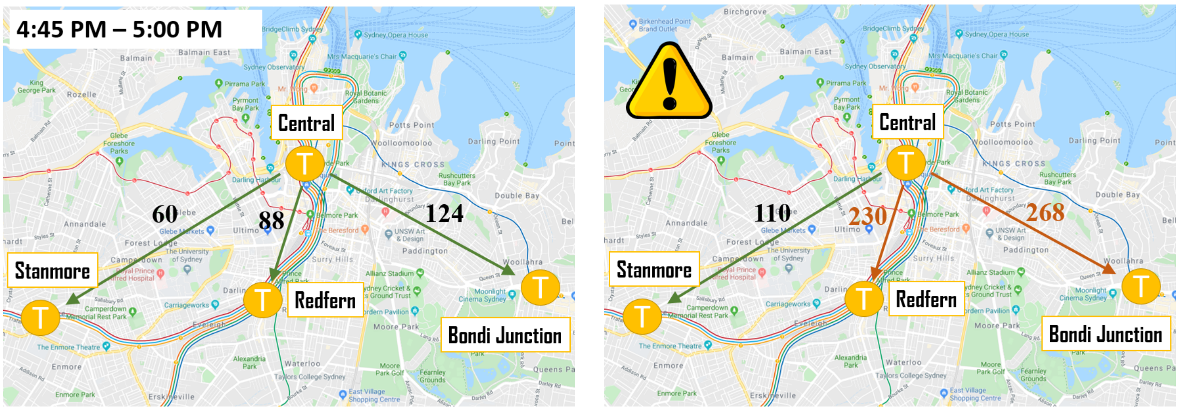
4.1 Introduction

Crowd flow prediction is not only one of the crucial research hotspots in the field of intelligent transportation system and urban computing, but is also recognized as an important real-world application benefiting metro development and urban services, such as risk assessment, route planning, congestion avoidance, etc. [45, 51, 55, 103]. It is very important for public safety: for instance, streamed people caused a chaotic crowd stampede at the Falls Festival in Lorne on Victoria's south-west coast, leaved up to 80 people injured; and 36 people died in a catastrophic stampede at the 2015 New Year's Eve celebrations in Shanghai [97]. An effective crowd warning and prediction system can effectively prevent people from such real tragedies by utilizing emergency mechanisms.

Thanks to the modern smart-card ticketing system for travel on public transport and portable GPS devices, a large collection of transactional data with spatiotemporal information are now available for analysts to make full use of data and discover useful knowledge. At present, a number of successful and applicable crowd flow prediction methods have been proposed to improve operational performance of transit authorities[53, 59, 100]. Unfortunately, to the best of our knowledge, most existing methods focusing on metro crowd flow prediction are non-network-wide framework. They formulate their problems on separate stations or a few subway lines and none of these approaches can be implemented in forecasting CFD directly.

It should be noted, however, it is inadequate to concentrate solely on entrance and exit flows, managers also desperately want to solve the problem of getting potential passenger distributions, i.e., forecast the crowd flow distribution, termed as CFD forecasts in this chapter. Once the passenger flows across the entire transportation network can be gotten, the predicted CFD information will significantly assist the analysis of how a station affects others, which is utmost essential for urban transportation development, e.g., passenger route planning and train scheduling. For instance, Figure 4.1(a) presents a predicted snapshot. The model makes a forecast that there are 272 passengers departure from Central station between 4:45 PM and 5:00 PM. Among them, 124 passengers will arrive at Bondi Junction, 88 at Redfern, and 60 at Stanmore, respectively. Through obtaining the CFD forecasts among all metro stations, transport managers can timely forecast irregular flow patterns and make a global regulation to maintain the normal train scheduled and make a warning for crowd evacuation. Figure 4.1(b) illustrates this situation that when an irregular entrance flow appears in the Central station, the congestion warning will be transmitted to the all possibly affected stations (Redfern and Bondi Junction).

To date, limited techniques can be used directly to address the network-wide CFD prediction problem. Regression-based methods like Gaussian processes (GP) [107] and auto-regressive integrated moving averages (ARIMA) [84] are proposed to forecast



(a) 560 passengers departure from Central between 4:45 and 5:00 PM, and their distribution in the future.

(b) An effective crowd warning for all possibly affected stations when suffering from abnormal flow patterns.

Figure 4.1: An example of crowd flow distribution.

entrance and exit crowd flows. While other approaches, such as wavelet-SVM [70] and probability trees [36] have successfully designed address the classical crowd flow prediction problem, they are hard to implement into the entire metro network. Even though deep neural networks [83, 96, 97], are able to fix the network-wide crowd flow prediction problem, they are sensitive to parameters and incomplete inputs, and require large training data that are not in line with our task. To summarize, our CFD prediction problem faces with three intrinsic challenges here:

High computational complexity. The specific CFD prediction problem requires getting all potential flows across entire metro stations, which calculates the entrance/exit flows and CFD simultaneously. Most advanced models like [10, 12, 55, 70, 83], are already computationally expensive even on a few metro lines. Meanwhile, they require repeated large off-line training processes that are difficult to be applied in the online system and network-wide problem.

Dynamic complexity. The crowd flow changes dynamically which is influenced by complicating factors, such as time, station similarity and climate conditions.

Real-time delayed data collection. Considering the online system, when we focus on entrance CFD prediction, there is a travel time gap between a passenger enters a station and exits another. These time gaps lead to the online system cannot collect complete data because there are a large number of passengers still on their journeys. In this situation, most city-wide traffic flow prediction methods, such as [23, 96–98], fail to solve our problem because they require the complete data in training and testing processes. Section 4.2.2 illustrates the detailed discussions and explanations.

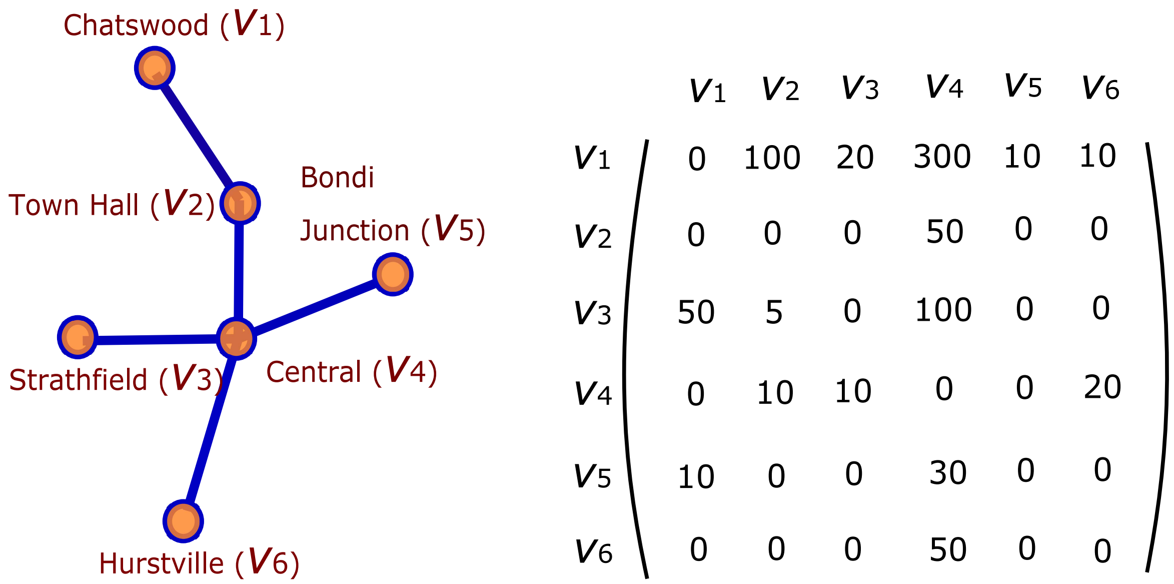
Motivated by these challenges, we propose three online latent space learning models for the CFD prediction of complex metro network, which enables us to forecast both CFD and entrance/exit flows precisely. Recently, the online latent space strategy has been used in the traffic flow prediction task and recognized as an advanced approach to address the spatiotemporal network-wide problem [4, 16, 22]. These models, firstly embedded the network-wide data into latent spaces by utilizing matrix factorization based methods. Then, temporal information is also involved to capture latent attributes and detect the dynamic patterns with time evolves [8].

In Chapter 4, to effectively infer the spatiotemporal latent attributes among different stations, we take advantages of the non-negative matrix factorization (NMF) strategy to embed the CFD network of each timestamp into two latent spaces; the first latent space indicates the properties of all entrance metro stations, and the second one indicates the properties of all exit metro stations. The first contribution of this chapter is to propose a CFD prediction model, called OLS-AO (online latent space model with average optimization), which is able to learn a smooth tendency by utilizing an average optimization strategy from previews timestamps in a given time window. Global similarities among all stations and climates are also involved in this model by a graph Laplacian embedding approach. However, the sudden increase/decrease flows will appear when suffering from some irregular events or during peak times. In such scenarios, the average strategy may prevent OLS-AO from gaining the sudden changes in flows, misleading to the next prediction. To keep the effectiveness of our method running in real-world applications, our second contribution is that we further design another variant model via the optimization of the most recent CFD trends, termed as OLS-MR (online latent space model with most recent trend). Empirical results demonstrate that the second model, OLS-MR, achieves better performances than OLS-AO when crowd flows change dramatically. Accordingly, given that each model performs relatively better in various situations, our third contribution is to propose a dual-track model, called OLS-DT, that takes advantages from both OLS-AO and OLS-MR in a parallel running way. We perform a collection of experiments on a large real-world transactional data covering Sydney Trains and make comparisons with other state-of-the-art methods to test the effectiveness of our approaches. These are: (1) CFD predictions across the entire network; (2) major station test; (3) comparisons between weekdays and weekends; rush and non-rush hours.

It should be noted that though our models are proposed to address the CFD prediction problem, they can be transfered to other network-wide crowd flow prediction problem, such as forecasting city-wide crowd flows [97]. To demonstrate the transferability of

Table 4.1: Symbol description.

Symbols	Descriptions
$G; n$	city trains network; number of stations
X_t	entrance or exit CFD matrices
$W; H$	latent space matrices
k	the number of dimensions of latent attributes
T	the number of consecutive snapshots in a window
$A; B$	transition matrices of W and H
P	indication matrix for all complete entries of D
$\lambda_1 - \lambda_3; \hat{\lambda}_1 - \hat{\lambda}_3$	regularization parameters



(a) A sample network G.

$$\begin{array}{c|cccccc} & V1 & V2 & V3 & V4 & V5 & V6 \\ \hline V1 & 0 & 100 & 20 & 300 & 10 & 10 \\ V2 & 0 & 0 & 0 & 50 & 0 & 0 \\ V3 & 50 & 5 & 0 & 100 & 0 & 0 \\ V4 & 0 & 10 & 10 & 0 & 0 & 20 \\ V5 & 10 & 0 & 0 & 30 & 0 & 0 \\ V6 & 0 & 0 & 0 & 50 & 0 & 0 \end{array}$$

(b) X_t $(\boldsymbol{x}_{ij})$

Figure 4.2: The topology example of metro network.

our methods, we choose another dataset used in [97] to test our performances. All the evaluation results prove the superiority of our models over comparison methods.

4.2 Problem Description

In this section, we first describe the crowd flow related data, and then formulate the network-wide CFD prediction problems as a graph network problem. For ease of presentation, the main symbols used in this chapter are summarized in Table 4.1.

4.2.1 Data Description

We describe the mainly used dataset in this chapter, which is a large-scale, real-world transactional dataset of Sydney Trains network. After data cleaning[1], the dataset contains above 30 million transactional records covering 178 stations between 7 Nov 2016 and 11 Dec 2016. We also downloaded the weather data from Australia Bureau of Meteorology[2] of the same time span. Below sections introduce their formats and features.

4.2.1.1 Transactional data

Transactional database stores a large number of activity attributes of passengers. We only used the records that are related to our problem and fulfilled the confidentiality deed that ensures the privacy of each passenger. The useful records include: **(1) Passenger**: a unique hashed identification number; **(2) Origin**: the time and location where the passenger started a journey; **(3) Destination**: the time and location where the passenger ended a journey; **(4) Duration**: the number of seconds taken to complete a journey. An example of data is presented in Figure 4.3 (a).

4.2.1.2 Station throughput

The station throughput data record the throughputs (entrance and exit) of each station at all timestamps (15 minute interval). This database can reflect the busy degree of each station. An example of data is shown in Figure 4.3 (b).

4.2.1.3 Weather

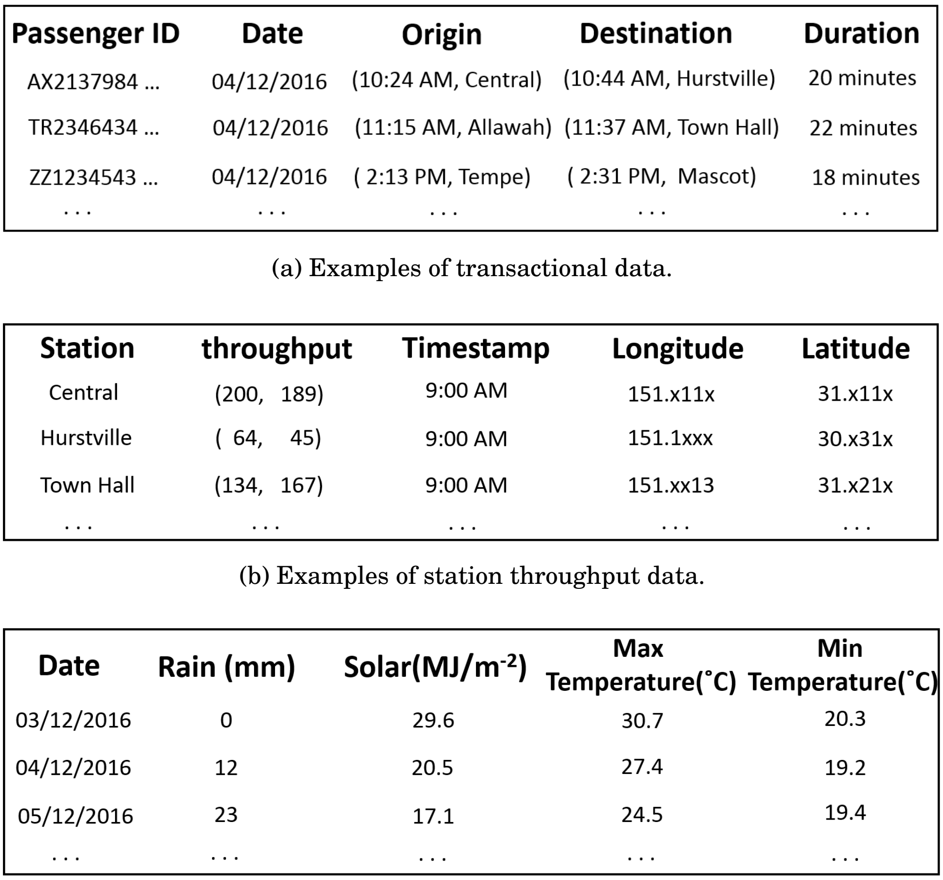
Weather data are collected from the Australia Bureau of Meteorology. Our data are gathered from the longitude and latitude of stations in the entire Sydney area. In this case, the weather conditions for them may vary. Figure 4.3 (c) indicates an example of our weather data.

4.2.2 Problem Formulation

Focusing on the CFD prediction problem, we need to record every travel path of every passenger. In this chapter, each travel path is termed as an origin-destination pair. We use a directed graph $G = (V, E)$ to define the CFD network, where E is the group of edges

[1] We removed the recording errors, UNKNOWN trips, and entrance and exit at the same station, etc.

[2] www.bom.gov.au/climate/data/



Passenger ID	Date	Origin	Destination	Duration
AX2137984 ...	04/12/2016	(10:24 AM, Central)	(10:44 AM, Hurstville)	20 minutes
TR2346434 ...	04/12/2016	(11:15 AM, Allawah)	(11:37 AM, Town Hall)	22 minutes
ZZ1234543 ...	04/12/2016	(2:13 PM, Tempe)	(2:31 PM, Mascot)	18 minutes
...	...	...	...	...

(a) Examples of transactional data.

Station	throughput	Timestamp	Longitude	Latitude
Central	(200, 189)	9:00 AM	151.x11x	31.x11x
Hurstville	(64, 45)	9:00 AM	151.1xxx	30.x31x
Town Hall	(134, 167)	9:00 AM	151.xx13	31.x21x
...	...	...	...	...

(b) Examples of station throughput data.

Date	Rain (mm)	Solar(MJ/m^{-2})	Max Temperature(˚C)	Min Temperature(˚C)
03/12/2016	0	29.6	30.7	20.3
04/12/2016	12	20.5	27.4	19.2
05/12/2016	23	17.1	24.5	19.4
...	...	...	...	...

(c) Examples of weather data.

Figure 4.3: A sample of our data.

and V denotes the group of vertexes. A vertex $v_i \subseteq V$ records the i-th entrance metro station, and an edge $e(v_i,v_j)$ denotes an origin-destination pair from station v_i to v_j. Assume that the current time is T, for any previous time interval t, the value of each edge $e(v_i,v_j)$ is associated with the observed flow $x(v_i,v_j)$, i.e., $x(v_i,v_j)$ is the total number of passengers that departure from i^{th} station at timestamp t and are going to the j^{th} station. We collect and calculate these passenger numbers from the travel information in the transactional data, i.e., if a passenger is detected departure from the i^{th} station at t, and has arrived at j^{th} station at current time T, the number will add one in the corresponding position of the CFD matrix. Then, G can be represented by CFD matrix $X_t = (x_{ij})$, where $x_{ij} = x(v_i,v_j)$.

The time interval is 15 minutes, which is a appropriate and practical timespan in the real-world application [12, 70, 78]. With time evolves, there are different CFD matrix X_t at timestamp t. For instance, a sample of real Sydney trains network G_t is shown in

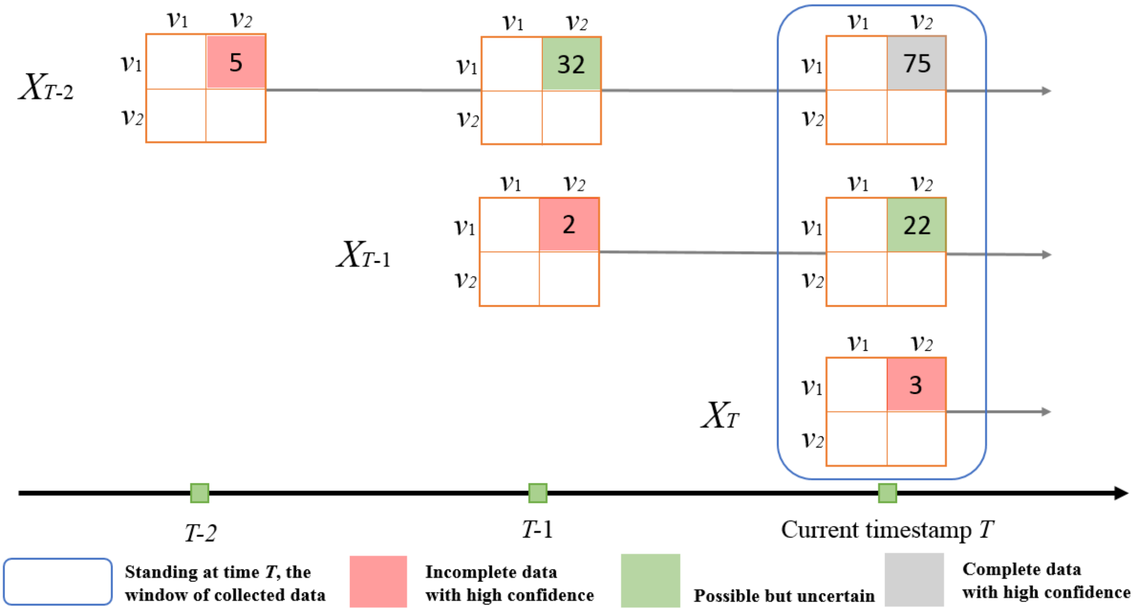

X_{T-2}
v_1 v_2
5
32
75
X_{T-1}
2
22
X_T
3
T-2
T-1
Current timestamp T
Standing at time T, the window of collected data
Incomplete data with high confidence
Possible but uncertain
Complete data with high confidence

Figure 4.4: An example of delayed data collection. Suppose there are two stations (v_1 and v_2), and we will only focus on the OD pair from v_1 to v_2. At the current timestamp T, the data in X_T and X_{T-1} are increasing until all passengers have reached their destinations. The blue box illustrates the data we can collect at T. Can we use the collected data “3” in X_T as a complete data? No, because there are a large number of passengers still on their journeys. Does “22” indicate the complete number of travels in X_{T-1}? Possible but uncertain, because there are many routes (or express and local train) between v_1 and v_2, the faster one may have arrived in one time interval, but the slower one maybe not. Make our attention at X_{T-2}. Is the number “75” complete? Much more possible, because two time intervals passed.

Figure 4.2(a), its corresponding CFD matrix X_t is shown in Figures 4.2(b), where x_{12} = 100 means that we detected 100 travelers entering at Chatswood (v_1) at time interval t and have exited at Town Hall (v_2) at the current time interval T.

Real-time delayed data collection problem. Standing at the current timestamp T, it is infeasible to build an exact X_T (i.e., an exact X_T is that all passengers have arrived their destinations). It is because X_T needs to collect all crowd flows, the value, such as x_{ij}, is keep growing in the next several timestamps until all passengers have reached their destination. Figure 4.4 illustrates an example of how the delayed data affect the data collection. In this case, how can we make sure that the data collected are complete and can be used for learning? To address this challenge, we propose a complete data condition in Section 4.3.2.2.

In summary, we aim to predict the next short-term CFD matrix X_{T+1} on-the-fly by

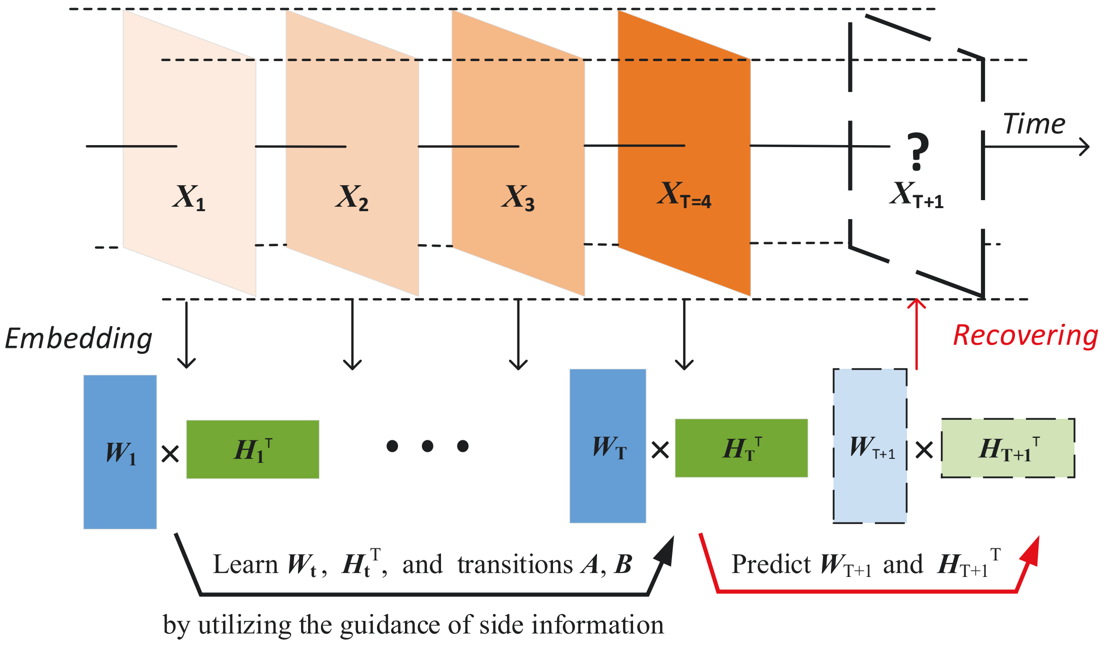

Time
?
X_1
X_2
X_3
$X_{T=4}$
X_{T+1}
Embedding
Recovering
W_1 × H_1^T
• • •
W_T × H_T^T
W_{T+1} × H_{T+1}^T
Learn W_t, H_t^T, and transitions A, B
Predict W_{T+1} and H_{T+1}^T
by utilizing the guidance of side information

Figure 4.5: The flowchart of OLS-AO. In the learning process, given a set of previous CFD matrices $\{X_t\}$ with the time window T (use $T = 4$ as an example), OLS-AO learns the latent spaces W_t and H_t of each X_t and the transitions matrices A and B by an average optimization method in section 4.3.2.3. The side information is utilized to guide the updating of W_t, H_t, A and B during the learning process. Predicted latent spaces W_{T+1} and H_{T+1} can be inferred by the Algorithm 2 shown in section 4.3.4.

using a series of previous CFD, X $(X_1, X_2, \cdots, X_T)$.

4.2.3 Exit Crowd Flow Prediction Problem

In the above section, we discussed the origin to destination (OD) flows, termed as entrance CFD. However, in real-world scenarios, there is another part of CFD, named as exit CFD. For example, let $y(v_i, v_j)$ records the number of people that exit at i^{th} station and came from j^{th} station (i.e., a destination to origin pair, (DO)). The value (v_3, v_1) = 50 shown in Figure 4.2(b) illustrates that there are 50 passengers exit at Strathfield (v_3) at time interval t, that came from Chatwood (v_1). In this situation, we can construct an exact exit CFD matrix because all the passengers have arrived at their destinations. Although there are two CFD types, they can be addressed in a similar way. Thus, to keep clarity of this chapter, we have only presented the optimization strategy for entrance CFD prediction, but test them both in the experiments.

4.3 Online Latent Space Model: OLS-AO

In this section, we propose our first model OLS-AO (online latent space with average optimization). We will describe how the latent space model can be used in the metro network, and how to make the model capture temporal patterns. OLS-AO is able to learn the latent temporal transitions via an average optimization strategy, which takes into account climate influences and similarities among different stations as well as the historic trends. Figure 4.5 shows the flowchart of this model.

4.3.1 The Basic Latent Space Model

Considering a slice X at timestamp t, the basic latent space model decomposes the CFD matrix $X \in \mathbb{R}_{+}^{n \times n}$ into two matrices $W \in \mathbb{R}_{+}^{n \times k}$ and $H \in \mathbb{R}_{+}^{n \times k}$, where W and H represent the latent spaces; n is the number of stations and k is the number of dimension of latent space. Each row in these matrices represents k attributes of corresponding entrance and exit stations. Hence, the crowd flows between the entrance and exit stations are determined by the interactions between latent attributes.

$$\min_{W \geq 0, H \geq 0} ||X - WH^{\top}||_F^2, \tag{4.1}$$

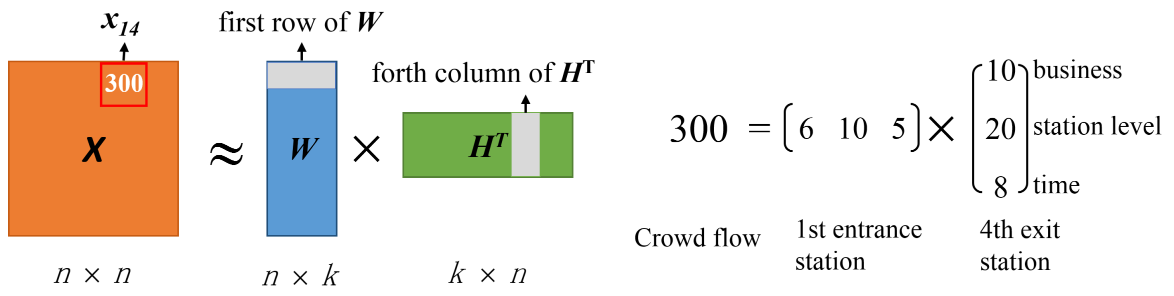
where $H^{\top}$ is the transpose of H.

In this basic latent space model, we utilize the non-negative matrix factorization method. One of the advantages of non-negative constraint is the interpretability of the results and reasonable assumptions of latent attributes [35, 38, 47]. Besides, due to the fact that the predicted values must be non-negative, W and H should be non-negative as well. Figure 4.6 gives the intuition of this basic model.

4.3.2 Online Strategy

4.3.2.1 Temporal information involved

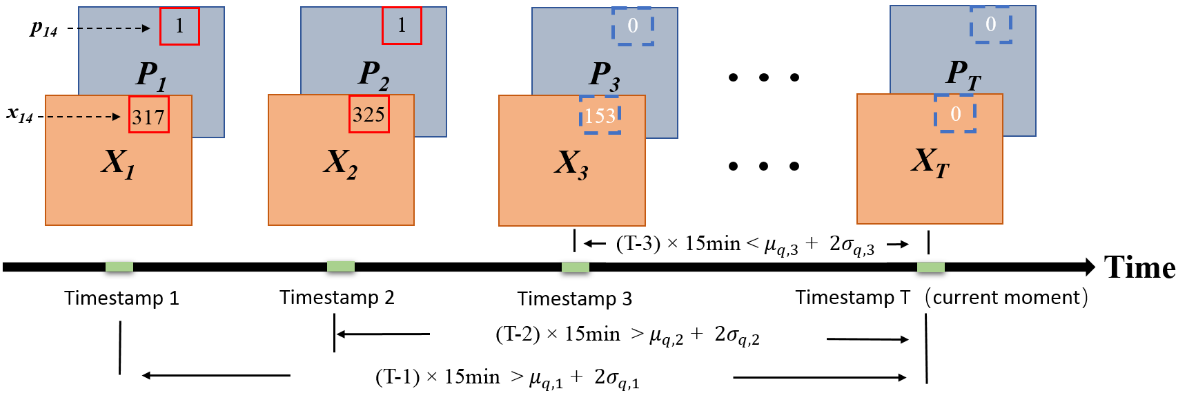
To involve the temporal information, we formulate the original continuous problem as a time-dependent model. Given a time window T (i.e., a sequence contains T previous CFD matrix, and T is also used to present the current time), for each timestamp $t \in (1, \cdots, T)$, we aim to learn the corresponding time-dependent latent attribute representations W_t and H_t from X_t. After involving the time dimension, our model can be expressed as:



(a) CFD matrix decomposition.

(b) crowd flow estimation.

Figure 4.6: The latent space example. It represents how to build the static latent space model for our CFD problem in each timestamp. As shown in subfigure(a), crowd flow (x_{14}) is determined by two sets of latent attributes. These attributes might illustrate many factors, such as time spans, business region, station size, etc. It is remarkable that subfigure(b) provides an example for these latent attributes when $k = 3$, and these latent attributes can be any factors without existing a strict explanation. The dimension of latent space k is a hyper-parameter.



p_{14}
1
P_1
x_{14}
317
X_1
1
P_2
325
X_2
0
P_3
153
X_3
0
P_T
0
X_T
(T-3) × 15min < $\mu_{q,3}$ + $2\sigma_{q,3}$
Time
Timestamp 1
Timestamp 2
Timestamp 3
Timestamp T (current moment)
(T-2) × 15min > $\mu_{q,2}$ + $2\sigma_{q,2}$
(T-1) × 15min > $\mu_{q,1}$ + $2\sigma_{q,1}$

Figure 4.7: An example of building indication matrix P_t. We take the entries x_{14} and p_{14} as the example. If the values meet the condition of data completion, then we can use these values as the guidance, $p_{14} = 1$ as shown in the red solid line box; if not, set p_{14} to 0 which means the collected data are incomplete yet as shown in the blue dotted line box.

$$\min_{W_t \geq 0, H_t \geq 0} \mathcal{J} = \sum_{t=1}^{T} ||P_t \odot (X_t - W_t H_t^\top)||_F^2, \tag{4.2}$$

where $\odot$ is the entrywise product; and P_t denotes the indication matrix for all the complete trips in X_t. The construction process of P_t is discussed next.

4.3.2.2 Address the delayed data collection problem

For entrance CFD problem, we only set $P_t = 1$ $(1 \le i, j \le n)$ if the time horizon between t and present time T is sufficient for the vast majority of people to have arrived at their destination. As we mentioned in Section 4.1 and 4.2, the real-time delayed data collection problem illustrates that we can only obtain a part of complete trips, and other flows are incomplete until the last passenger finish his journey and exit at station. In this case, how can we make sure that the data collected are complete?

Complete data condition. To make the problem solvable, we assume that the travel times for each OD pair in each timestamp fits the normal distribution based on the suggestion in [27], i.e., $Z_{q,t} \sim N(\mu_{q,t}, \sigma^2_{q,t})$. $Z_{q,t}$ denotes the travel times for one OD pair at timestamp t (q^{th}, $q \in$ all CFD). Then, based on the property of normal distribution, if the time horizon is greater than $\mu_{q,t} + 2\sigma_{q,t}$, we have approximately 98% confidence that all passengers have reached their destination.

Figure 4.7 represents the process of constructing the indication matrix P_t. X_1 is the first CFD matrix in the time window. The value of x_{14} means we have collected 317 passengers that enter at v_1 at time t=1 and then exit at v_4 until T. Due to the fact that the time gap between 1 and T is larger than the learned $\mu_{q,1} + 2\sigma_{q,1}$, we have a very high confidence that the collected data (317) is complete. In contrast, as shown in X_3, the time gap between 3 and T is smaller than $\mu_{q,3} + 2\sigma_{q,3}$, which does not gain the enough confidence level. Then the entry 153 is recognized as a incomplete data.

4.3.2.3 Latent Transition Learning

Since our model is an online dynamic prediction system, crowd flow is continually changing over time. We focus on learning the transition patterns of latent spaces W_t and H_t from previous timestamps to the next. The evolving patterns can be captured by these learned transitions, so we can do a prediction based on the current CFD condition.

In our first model, OLS-AO, we choose an average optimization strategy to learn two transition matrices $A \in \mathbb{R}_+^{k \times k}$ and $B \in \mathbb{R}_+^{k \times k}$, which represent the smooth trends of W_t and $H_t^\top$ in T previous CFD matrices. The average optimization strategy is able to filter accidental noise in some degrees, such as urgent construction, Gate failure, etc. Therefore, the learned trends can be recognized as a representation of the stable CFD changes from previous timestamps. For example, A and B approximate the changes of W and H between t-1 to t, i.e., $W_t = W_{t-1}A$ and $H_t = H_{t-1}B$.

To this stage, the latent transition learning process is shown as Equation (4.4).

$$\min_{W_t,H_t,A,B} \mathscr{T} = \sum_{t=2}^{T}(||W_t - W_{t-1}A||_F^2 + ||H_t - H_{t-1}B||_F^2). \tag{4.3}$$

We consider our online optimization problems jointly, then we can get:

$$\min_{W_t,H_t,A,B} \mathscr{L}_1 = \mathscr{I} + \lambda_1 \mathscr{T}, \tag{4.4}$$

where λ_1 is the regularization parameter.

In a real-world application, crowd flows would be influenced by various complex factors, such as the similarities among stations, weather, and periodic property. Therefore, to improve the forecasting performance, we will take these valuable resources into consideration in the next section.

4.3.3 Learning From Side Information

4.3.3.1 Incorporating with Inherent and External Influences

Based on the phenomenon that the development of a city gradually evolves various functional regions, such as residential, business, and tourist areas, where the regions have the same functional property will have strong connections with each other [103]. The stations located in the same functional regions will also have strong similarities with each other. To consider the inherent features and extract much more non-linear information, we build the kernel K_t^s to present the correlations among stations.

Let $\mathbf{x}_i$ $(1 \le i \le n)$ be a flow distribution at timestamp t of i-th entrance station, and $\phi(\cdot)$ be the mapping that maps $\mathbf{x}_i$ onto the reproducing kernel Hilbert space. In this case, a kernel matrix K_t^s is then calculated by applying the kernel function $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$ into $\{\mathbf{x}_i\}_{i=1}^n$. The similarity kernel matrices are learned by historic data due to the incomplete data issue in the online process.

As similar to the building process of K_t^s, we can build a kernel matrix K_w based on the weather view. Unlike K_t^s changing with a small time span, K_w presents the weather condition in a day. We apply commonly used radial basis function (RBF) kernels to build K_t^s and K_w. For example, assume $\mathbf{r}_1$ and $\mathbf{r}_2$ indicate the weather properties of station v_1 and v_2 , the weather kernel K_w is generated by $K_w(\mathbf{r}_1, \mathbf{r}_2) = exp(-\delta||\mathbf{r}_1, \mathbf{r}_2||^2)$, where δ is an RBF kernel parameter, we set δ = 0.2 in our method. Then we use the following equation to fuse these two components.

$$K_t = K_t^s + \alpha Y_t \odot K_w, \tag{4.5}$$

where Y_t is a weight matrix at timestamp t, which is learned by the throughput data (normalized into [0, 1]); $Y_{t(i,j)}$ = 1 - $dist(\mathbf{y}_i, \mathbf{y}_j)$, where $dist()$ is the Euclidean distance; the function of Y_t is to balance the different influences between large and small stations; and α is a balance factor. The sensitivity analysis of all parameters and the effects of performance with various settings are shown in Section 4.5.9.

After above steps, we take into account the graph Laplacian dynamics [34] into our model to obtain the spatial structure of each station. Specifically, we construct a graph Laplacian matrix L_t, defined as $L_t = D_t - K_t$, where D_t is a diagonal matrix $D_{t(ii)} = \sum_j (K_{t(ij)})$. This new constraint can be expressed as the following optimization:

$$\min_{W_t} \quad \mathscr{S} = \sum_{t=1}^{T} Tr(W_t^\top L_t W_t). \tag{4.6}$$

The intuition behind this graph-based optimization is that one of the latent spaces W_t can learn the similarity structure from K_t.

4.3.3.2 Historical Guidance

Traffic periodicity is a very important factor for the existing studies. Crowd flows also represent the stable and daily periodic properties, especially on weekdays. Thanks to this phenomenon, our prediction objective X_{T+1} is close to its historic guidance X_{T+1}^h. Thus, history information is an important guidance that should be considered. We denote that W_{T+1}^h and H_{T+1}^h are two latent matrices learned from history for the next timestamp T+1. We aim to learn variables by consulting these historical guidances. More importantly, the other benefit of this strategy is that it adapts to the sharpening transformation from peak time to non-peak time.

The historical guidance W_{T+1}^h and H_{T+1}^h can be learned by Equation (4.4) with a slight modification, of replacing X_t, W_t and H_t with X_t^h, W_t^h and H_t^h, respectively, and setting T to $T+1$. The indication matrix P_t should be removed in the learning process because all historical trips are completed. Then, historical guidance is expressed as:

$$\min_{W_T, H_T, A, B} \mathscr{H} = ||W_{T+1}^h - W_T A||_F^2 + ||H_{T+1}^h - H_T B||_F^2, \tag{4.7}$$

where W_T and H_T are the latent spaces at timestamp T.

Taking all the above techniques into consideration, our final jointly loss function is expressed as:

$$\min_{W_t, H_t, A, B} \mathscr{L} = \mathscr{I} + \lambda_1 \mathscr{T} + \lambda_2 \mathscr{S} + \lambda_3 \mathscr{H}, \tag{4.8}$$

where λ_1 to λ_3 are the regularization parameters.

After solving Equation (4.8), the learned matrices W_T, H_T, A and B can be used to do the prediction. The next CFD matrix X_{T+1} is:

$$X_{T+1} = (W_T A)(H_T B)^\top. \tag{4.9}$$

4.3.4 Learning Process

Equation 4.8 is a complex non-convex problem. In this case, we choose an effective gradient descent method with multiplicative update strategy [35] to discover the local optimization.

Theorem 4.1. *$\mathscr{L}$ is non-increasing under the following update rules in Equation (4.10)-(4.14) by optimizing W_t, H_t, A and B alternatively:*

$$W_t = W_t \odot \frac{P_t \odot X_t H_t + \lambda_1 (W_{t-1} A + W_{t+1} A^\top) + \lambda_2 K_t W_t + \hat{\lambda_3} W^h_{T+1} A^\top}{P_t \odot (W_t H_t^\top) H_t + \lambda_1 (W_t + W_t A A^\top) + \lambda_2 D_t W_t + \hat{\lambda_3} W_T A A^\top}, \tag{4.10}$$

$$H_t = H_t \odot \frac{P_t^\top \odot X_t^\top W_t + \lambda_1 (H_{t-1} B + H_{t+1} B^\top) + \hat{\lambda_3} H^h_{T+1} B^\top}{P_t^\top \odot (H_t W_t^\top) W_t + \lambda_1 (H_t + H_t B B^\top) + \hat{\lambda_3} H_T B B^\top}, \tag{4.11}$$

where $\hat{\lambda_3}$ is given by:

$$\hat{\lambda_3} = \begin{cases} \lambda_3, \, t = T \\ 0, \, otherwise \end{cases} \tag{4.12}$$

$$A = A \odot \frac{\lambda_1 \sum_{t=1}^{T} W_{t-1}^\top W_t + \lambda_3 W_T^\top W^h_{T+1}}{\lambda_1 \sum_{t=1}^{T} W_{t-1}^\top (W_{t-1} A) + \lambda_3 W_T^\top (W_T A)}, \tag{4.13}$$

$$B = B \odot \frac{\lambda_1 \sum_{t=1}^{T} H_{t-1}^{\top} H_t + \lambda_3 H_T^{\top} H_{T+1}^{h}}{\lambda_1 \sum_{t=1}^{T} H_{t-1}^{\top}(H_{t-1}B) + \lambda_3 H_T^{\top}(H_T B)}. \tag{4.14}$$

It is noteworthy that the update rules indicate a time-related learning process, the information from previous and next timestamps (W_{t-1} and W_{t+1}) affect each other. This chain updating rules make sure all variables are learned comprehensively with the side information guidance. We now derive the update rule of W_t as an example, other variables can be solved with a similar process. Considering W_t while all other variables are fixed, Equation (4.8) can be rewritten as follows:

$$\begin{aligned} \mathscr{L} = &\sum_{t=1}^{T} Tr((P_t \odot (X_t - W_t H_t^{\top}))(P_t \odot (X_t - W_t H_t^{\top}))^{\top}) + \\ &\lambda_1 \sum_{t=2}^{T} Tr((W_t - W_{t-1}A)(W_t - W_{t-1}A)^{\top}) + \lambda_2 \sum_{t=1}^{T} Tr(W_t^{\top} L_t W_t) \\ &+ \lambda_3 Tr((W_{T+1}^{h} - W_T A)(W_{T+1}^{h} - W_T A)^{\top}) + \mathscr{J}(H_t, B), \end{aligned} \tag{4.15}$$

where $\mathscr{J}(H_t,B)$ indicates the components of $\mathscr{L}$ excluding W_t, which is a constant value when only considering the partial derivative of W_t.

Taking the derivation of $\mathscr{L}$ with respect to W_t, we can get $g(W_t)$ according to [60]:

$$\begin{aligned} g(W_t) = &2(-P_t \odot X_t H_t + P_t \odot (W_t H_t^{\top}) H_t) + \\ &2\lambda_1(-W_{t-1}A + W_t - W_{t+1}A^{\top} + W_t A A^{\top}) + \\ &2\lambda_2(-K_t W_t + D_t W_t) + 2\hat{\lambda}_3(-W_{T+1}^{h} A^{\top} + W_T A A^{\top}), \end{aligned} \tag{4.16}$$

As introduced in [35], the traditional gradient descent method is expressed as: W_t = W_t - $\gamma g(W_t)$ = W_t - $\gamma(P_{item} + N_{item})$, where P_{item} and N_{item} denote all positive and negative items in $g(W_t)$, respectively (e.g., P_{item} = $2(P_t \odot (W_t H_t^{\top}) H_t + \lambda_1(W_t + W_t A A^{\top}) + \lambda_2 D_t W_t + \hat{\lambda}_3 W_T A A^{\top})$). We can set the step γ to:

$$\gamma = \frac{W_t}{P_{item}}, \tag{4.17}$$

then, we got the update rule of W_t as shown in Equation (4.10).

Algorithm 2 summarizes our learning and prediction process of OLS-AO.

```
Algorithm 2: OLS-AO
Input: CFD matrices $[X_1, \cdots, X_T]$; similarty kernels $[K_1, \cdots, K_T]$; historic
       guidance $W^h_{T+1}$, $H^h_{T+1}$
Output: prediction $X_{T+1}$
initialize $W_t, H_t, A$ and $B$ by historical values
for Epoch = 1 to M do
    if $|\mathscr{L}_i - \mathscr{L}_{i+1}| \;/\; \mathscr{L}_i \geq \varepsilon$ then
        for $t$ = 1 to $T$ do
            update $W_t$ and $H_t$ By Equation (4.10) - (4.12)
        update $A$ and $B$ By Equation (4.13) and (4.14)
    else
        Break
Return $X_{T+1}$ By Equation (4.9)
```

4.3.5 Analysis of Complexity and Convergence

We first discuss the computational complexity of OLS-AO. As shown in Equations (4.10) - (4.14) and Algorithm 2, the computational complexity is dominated by the matrix multiplication operations. For each iteration, the computational complexity of OLS-AO is $O(Tn^2k)$.

In terms of convergence, Algorithm 2 is guaranteed to converge at a local optimal solution by optimizing W_t, H_t, A and B alternatively. The convergence trends are plotted in Section 4.5.10. To prove **Lemma 4.1**, we will find an auxiliary function similar to that used in the [35]. We here give the convergence proof of H_T and other variables can similarly be proofed.

4.3.5.1 Proof of Theorem 4.1

Definition 1. *$\mathscr{Q}(h, h^{'})$ is an auxiliary function for our final function $\mathscr{L}(h)$ if the following conditions are satisfied:*

$$\mathscr{Q}\left(h', h\right) \geq \mathscr{L}(h) \quad \text{and} \quad \mathscr{Q}(h, h) = \mathscr{L}(h). \tag{4.18}$$

Lemma 4.1. *If $\mathscr{Q}$ is an auxiliary function, then $\mathscr{L}$ is non-increasing under the update:*

$$h^{t+1} = \arg\min_h \mathscr{Q}\left(h, h^t\right), \tag{4.19}$$

consequently, we have:

$$\mathscr{L}\left(h^{t+1}\right) \leq \mathscr{Q}\left(h^{t+1}, h^{t}\right) \leq \mathscr{Q}\left(h^{t}, h^{t}\right) = \mathscr{L}\left(h^{t}\right). \tag{4.20}$$

The proof of Lemma 4.1 is given by [35]. Lemma 4.1 illustrates that $\mathscr{L}\left(h^{t+1}\right) \leq \mathscr{L}(h^t)$ when exits $\mathscr{Q}\left(h, h^t\right)$.

Lemma 4.2. *If $R(h^t)$ is a diagonal matrix under the following definition,*

$$R(h^t) = diag((W^T diag(p) W + I + B^\top B)h/h), \tag{4.21}$$

where p is a column vector of $P_T \odot P_T$, then,

$$\mathscr{Q}\left(h, h^t\right) = \mathscr{L}\left(h^t\right) + \left(h - h^t\right)^T \nabla \mathscr{L}\left(h^t\right) + \frac{1}{2}\left(h - h^t\right)^T R\left(h^t\right)\left(h - h^t\right), \tag{4.22}$$

is an auxiliary function for $\mathscr{L}$.

Proof. Since $\mathscr{Q}(h,h) = \mathscr{L}(h)$ is obvious, we need only show that $\mathscr{Q}(h,h^t) \geq \mathscr{L}(h)$, which is equivalent to

$$0 \leq \left(h - h^t\right)^T \left[R\left(h^t\right) - (W^T diag(p) W + I + B^\top B)\right]\left(h - h^t\right). \tag{4.23}$$

Let O = $W^T diag(p) W + I + B^\top B$, then $\left[R\left(h^t\right) - (W^T diag(p) W + I + B^\top B)\right]$ can be expressed as $[diag(Oh/h) - O]$. As the proof provided in [24], if O is a symmetric non-negative matrix and h be a positive vector, then the matrix $\hat{O} = diag(Oh./h) - O \geq 0$. ■

Replacing $\mathscr{Q}(h,h^t)$ in Equation (4.19) by Equation (4.22), we can obtain update rules and proof $\mathscr{L}$ is non-increasing of Theorem 4.1.

4.4 A Variant Model OLS-MR and a Dual- Track Model OLS-DT

This section proposes one variant model OLS-MR (online latent space model via learning most recent trend) and a dual-track model to improve the CFD forecasting performance.

4.4.1 Motivation

Our first model, OLS-AO relies on the average optimization strategy to learn the transition matrices A and B. This learning method extracts long-term tendencies that can avoid abnormal noises, especially in the non-rush hours. However, it is insensitive to the sharp change of crowd flows. For example, Figure 4.8 (a) shows the real entrance flows of three stations between 10:00 AM and 3:00 PM. During these stable passenger flows, as shown in the green box, OLS-AO will perform well. However, a sudden increase flow will occur during the peak times, e.g., between 7:00 to 8:00 AM in Figure 4.8 (b), the crowd flow shoots upward. Intuitively, OLS-AO is insensitive to this scenario due to the average optimization strategy. Hence, this situation inspires us to propose another model to handle situations where sudden changes flow appear.

4.4.2 Learning the Most Recent Trend

As discussed above, we want to learn the most recent trend to fix the weakness of OLS-AO. Motivated by this point, we partition the transition matrices A and B by each timestamp t, i.e., learning A_t and B_t from one snapshot to the next. Note that, even though A and B are disassembled to the time-related variables, it does not mean that A_t or B_t only learn knowledge from timestamp t-1. There is an association chain between time 1 to time T, which can be illustrated by the update rules in Equations (4.27) - (4.30).

In detail, we revise the optimization function $\tilde{\mathscr{T}}$ and $\tilde{\mathscr{H}}$ in Equation (4.3) and (4.7) respectively to tackle the most recent dynamic trends as follows:

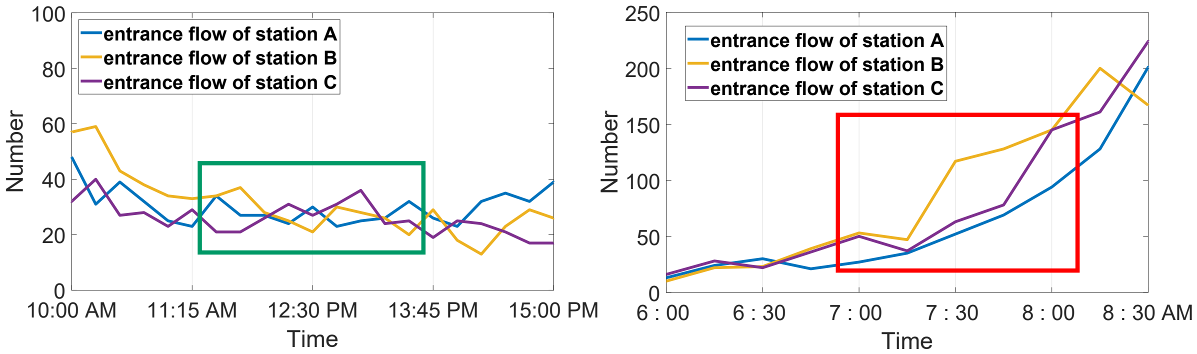
$$\min_{W_t,H_t,A_t,B_t} \tilde{\mathscr{T}} = \sum_{t=2}^{T}(||W_t - W_{t-1}A_{t-1}||_F^2 + ||H_t - H_{t-1}B_{t-1}||_F^2), \tag{4.24}$$

$$\underset{W_T,H_T,A_T,B_T}{min} \tilde{\mathscr{H}} = ||W_{T+1}^h - W_T A_T||_F^2 + ||H_{T+1} - H_T B_T||_F^2, \tag{4.25}$$

A_T and B_T are transition matrices when $t = T$. After incorporating the side information described in Section 4.3.3, the final loss function of OLS-MR is

$$\underset{W_t,H_t,A_t,B_t}{min} \tilde{\mathscr{L}} = \mathscr{I} + \eta_1\tilde{\mathscr{T}} + \eta_2\mathscr{S} + \eta_3\tilde{\mathscr{H}}, \tag{4.26}$$

where η_1 to η_3 are the regularization parameters.



(a) in non-rush hour.

(b) in rush hour.

Figure 4.8: Crowd flow changes in different scenarios.

4.4.3 Learning Process

Theorem 4.2. *$\tilde{\mathscr{L}}$ is non-increasing under the following update rules in Equation (4.27) - (4.30) by optimizing W_t, H_t, A_t and B_t alternatively:*

$$W_t = W_t \odot \frac{P_t \odot X_t H_t + \eta_1(W_{t-1}A_{t-1} + W_{t+1}A_t^\top) + \eta_2 K_t W_t + \hat{\eta_3} W_{T+1}^h A_T^\top}{P_t \odot (W_t H_t^\top) H_t + \eta_1(W_t + W_t A_t A_t^\top) + \eta_2 D_t W_t + \hat{\eta_3} W_T A_T A_T^\top}, \tag{4.27}$$

$$H_t = H_t \odot \frac{P_t^\top \odot X_t^\top W_t + \eta_1(H_{t-1}B_{t-1} + H_{t+1}B_t^\top) + \hat{\eta_3} H_{T+1}^h B_T^\top}{P_t^\top \odot (H_t W_t^\top) W_t + \eta_1(H_t + H_t B_t B_t^\top) + \hat{\eta_3} H_T B_T B_T^\top}, \tag{4.28}$$

$$A_t = A_t \odot \frac{\eta_1 W_t^\top W_{t+1} + \hat{\eta_3} W_T^\top W_{T+1}^h}{\eta_1 W_t^\top (W_t A_t) + \hat{\eta_3} W_T^\top (W_T A_T)}, \tag{4.29}$$

$$B_t = B_t \odot \frac{\eta_1 H_t^\top H_{t+1} + \hat{\eta_3} H_T^\top H_{T+1}^h}{\eta_1 H_t^\top (H_t B_t) + \hat{\eta_3} H_T^\top (H_T B_T)}. \tag{4.30}$$

Equations (4.27) - (4.30) satisfy:

$$\hat{\eta_3} = \begin{cases} \eta_3,\ t = T \\ 0,\ otherwise \end{cases}$$

Theorem 4.2 can be proved as shown in section 4.3.4 and 4.3.5 with slight changes.

Algorithm 3: OLS-MR

```
**Input:** CFD matrices $[X_1,\cdots,X_T]$; similarty kernels $[K_1,\cdots,K_T]$; historic
        guidance $W^h_{T+1}$, $H^h_{T+1}$
**Output:** prediction $X_{T+1}$
initialize $W_t$,$H_t$,$A_t$ and $B_t$ by historical values
**for** *Epoch = 1 to M* **do**
    **if** $\left|\tilde{\mathscr{L}}_i - \tilde{\mathscr{L}}_{i+1}\right| \ / \ \tilde{\mathscr{L}}_i \geq \varepsilon$ **then**
        **for** *t = 1 to T* **do**
            update $W_t$ and $H_t$ **By** Equations (4.27) and (4.28)
            update $A_t$ and $B_t$ **By** Equations (4.29) and (4.30)
    **else**
        Break
Return $X_{T+1}$ **By** $D_{T+1} = (W_T A_T)(H_T B_T)^\top$
```

Based on **Theorem 4.2**, the learning and prediction processes for OLS-MR are summarized in Algorithm 3.

We theoretically discuss the OLS-MR here. The computational complexity of OLS-MR is also determined by the matrix multiplication operations. For each iteration, the computational complexity is $O(Tn^2k)$. Furthermore, even though OLS-MR relies on the most previous trend, the learning process also leverage the information from previous and next timestamps (W_{t-1} and W_{t+1}), as well as the transition matrices. This chain updating rules make sure all training data are considered and associated with each other.

4.4.4 A Dual-track Model

As the fact that OLS-AO and OLS-MR execute respectively stronger in two different situations (the first adapts to the stable flows and the second adapts to the sudden changes of flows.) A dual-track strategy can be proposed to integrate both two models that to solve the prediction task in a variety of crowd flow situations.

Due to the *real-time delayed data collection* problem, the ground-truth at current time T is not available, but we only can obtain the total number of entrance crowd flows for each station at current time. Hence, we compare the sum of each row in X_T with ground-truth, and use MAE as the selection criteria for the next prediction. Inspired by [44], we use a temporally-varying fusion strategy in our dual-track model. The prediction of OLS-DT is expressed as:

$$X_{T+1}^{DT} = \begin{cases} \tau X_{T+1}^{AO} + (1-\tau) X_{T+1}^{MR} & if \ \ |MAE_{AO} - MAE_{MR}| \leq \xi \\ X_{T+1}^{AO} & if \ \ MAE_{MR} - MAE_{AO} > \xi \\ X_{T+1}^{MR} & if \ \ MAE_{AO} - MAE_{MR} > \xi \end{cases} \tag{4.31}$$

where $\tau = \frac{e^{-MAE_{AO}}}{e^{-MAE_{AO}} + e^{-MAE_{MR}}}$; and $\xi = 0.1$ is a threshold; MAE_{AO} and MAE_{MR} present the prediction errors at timestamp T of OLS-AO and OLS-MR, respectively. Our dual-track model (OLS-DT) is presented in Algorithm 4.

Algorithm 4: OLS-DT

```
while need to predict next $X_{T+1}$ do
    if $|MAE_{AO} - MAE_{MR}| \leq \xi$ then
        $X_{T+1}^{DT} = \tau X_{T+1}^{AO} + (1-\tau) X_{T+1}^{MR}$;
    else
        if $MAE_{MR} - MAE_{AO} > \xi$ then
            $X_{T+1}^{DT} = X_{T+1}^{AO}$;
        else
            $X_{T+1}^{DT} = X_{T+1}^{MR}$;
```

4.5 EXPERIMENTS

In this section, we report on the experiments carried out on three real-world datasets.

4.5.1 Datasets

• The state-wide train network. This dataset is provided by NSW Sydney Trains as shown in section 4.2.1. This dataset contains above 30 million trajectories covering 178 stations between 7 Nov 2016 and 11 Dec 2016; the average trajectories (including entrance and exit flows) on weekdays and weekends are 1.04 and 0.53 million, respectively. We pick the data between 7 Nov. 2016 and 20 Nov. 2016 as the training set (used to calculate the historic guidance and pick the parameters); the remaining data are used as the test set. For the detailed of strategy of picking the best hyper-parameters, please refer to the next section.

• The major station network. We select the top 20 stations as the major stations based on throughput capacity. This selected dataset contains This dataset contains about

19.8 million trajectories in total; the average trajectories (including entrance and exit flows) on weekdays and weekends are 0.68 and 0.34 million, respectively. The division of training and test data is the same as the first dataset.

• TaxiBJ. As we mentioned in the Introduction section, our methods not only can effectively address the CFD prediction, but also solve other network-wide flow forecasting task. We choose this widely used dataset to demonstrate the transferability of our models. For more details about this dataset, please refer to [97]. We used the data between 14 Mar. 2016 and 10 April. 2016 as our test data; the data between 15 Feb. 2016 and 13 Mar. 2016 were used as the training data (pick parameters).

4.5.2 Baselines & Measures & Parameters

Baselines. The baselines are outlined as follows. All parameters used in baselines are picked by a grid search approach.

• **HA:** We predict CFD by the historical average method on each timespan. For example, all historical time spans from 9:45 AM to 10:00 AM on Tuesdays are utilized to do the forecast for the same time interval.

• **ONMF:** the traditional Online non-negative matrix factorization based method without utilizing side information.

• **LSM-RN-All:** A state-of-the-art OLS-based method to predict network-wide traffic speed problem [16].

• **SARIMA:** A linear regression model with seasonal property to effectively predict future values in a time series.

• **GPR:** Gaussian process regression (GPR) would handle the spatiotemporal pattern prediction in a stochastic process. It usually suffers from a heavy computational cost [63].

Measures. Two metrics are used in this chapter, Mean Relative Error (MRE) and Mean Absolute Error (MAE), as they are generally employed in evaluating time series accuracy [93].

$$MAE = \frac{\sum_{i=1}^{\Omega} |c_i - \hat{c}_i|}{\Omega}, \qquad MRE = \frac{\sum_{i=1}^{\Omega} |c_i - \hat{c}_i|}{\sum_{i=1}^{\Omega} c_i},$$

where $\hat{c}_i$ is a forecasting value and c_i is the ground truth; Ω is the number of predictions.

Initialization. To ensure a better performance, we initialize latent space matrices of ONMF, LSM-RN-All, and proposed models in each time interval with corresponding

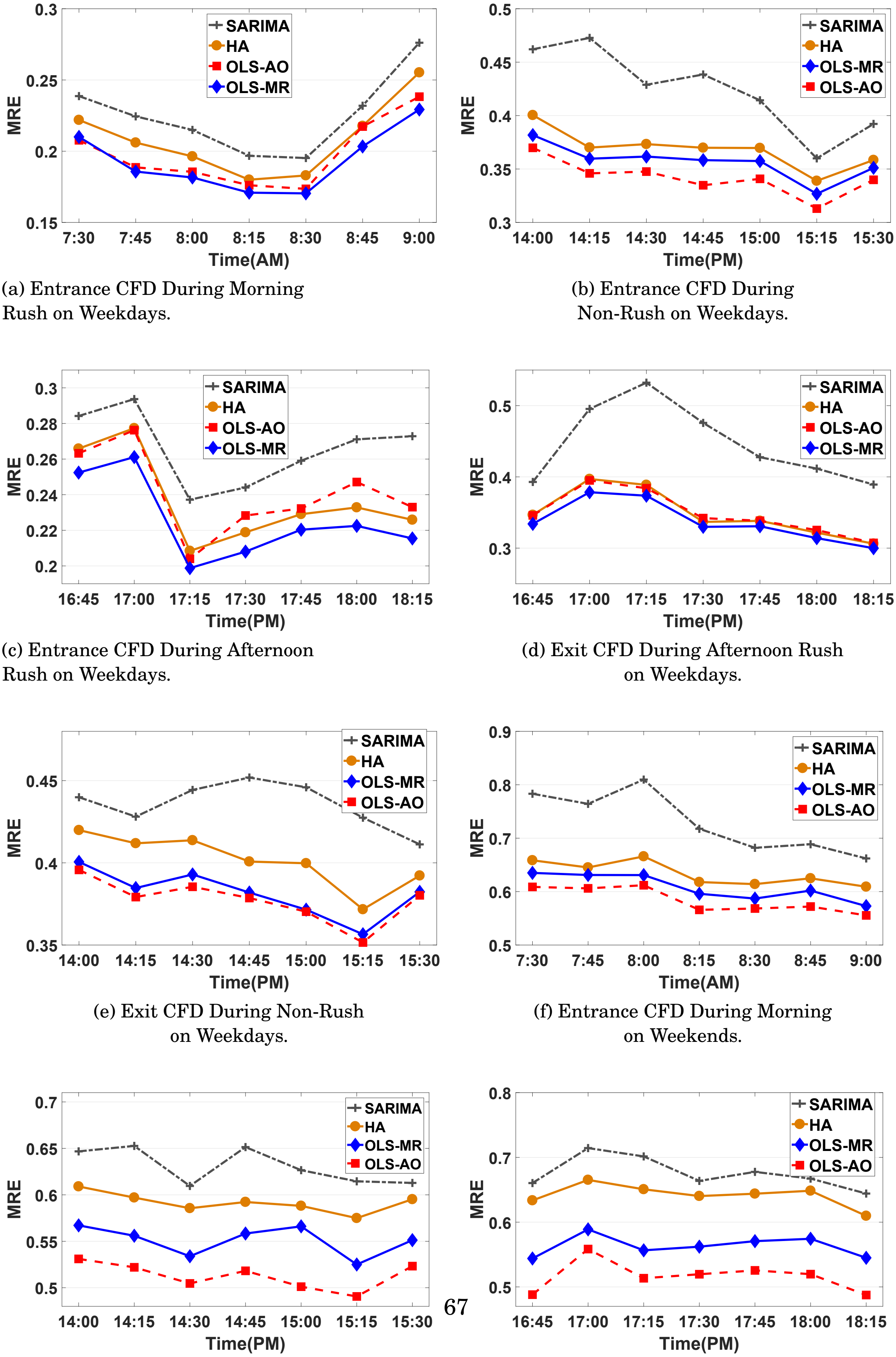
Table 4.2: Parameters.

Different types		OLS-AO			OLS-MR		
		λ_1	λ_2	λ_3	η_1	η_2	η_3
Weekdays	Entrance CFD	2^6	2^{-1}	2^{13}	$2^{1.5}$	2^{-5}	2^5
	Exit CFD	2^5	2^{-1}	2^{15}	$2^{5.5}$	2^{-4}	2^{18}
Weekends	Entrance CFD	2^6	2^{-3}	$2^{6.5}$	$2^{0.5}$	2^{-5}	2^2
	Exit CFD	2^2	2^{-2}	2^2	$2^{3.5}$	2^{-4}	2^4

historic data. This initialization method is reasonable under the assumption that model parameters will not change much compared to their historic values [16].

Picking Parameters. The hyper-parameters used in the experiments for different application scenarios are shown in Table 4.2. All regularization parameters are optimized by the grid search method. We set latent dimension $k = 70$, time window T of OLS-AO to 6 and T of OLS-MR to 3 for a good balance between time cost and performance. Much more details about sensitivity of parameters can be found in section 4.5.9.

It is worth notice that the proposed methods are online models. For example, when we want to predict CFD matrix at T+1, the data in T previous timestamps can be seen as the training set, and data at timestamp T+1 is used for validation. For the next prediction step, data at T+2 becomes the validation set and data in its T previous timestamps is recognized as the training set. Finally we go through all the training set, and get the average error on validation set. The hyper-parameters achieved the lowest error are picked in the test.



(a) Entrance CFD During Morning Rush on Weekdays.

(b) Entrance CFD During Non-Rush on Weekdays.

(c) Entrance CFD During Afternoon Rush on Weekdays.

(d) Exit CFD During Afternoon Rush on Weekdays.

(e) Exit CFD During Non-Rush on Weekdays.

(f) Entrance CFD During Morning on Weekends.

(g) Entrance CFD During Afternoon on Weekends.

(h) Exit CFD During Afternoon on Weekends.

Figure 4.9: CFD prediction on the entire trains network.

Table 4.3: Comparisons on different time spans. We report the average mean relative errors (MRE) through all test data and best results are bold. The time spans are M-rush (7:30-9:00 AM), Non-rush (14:00-15:30 PM), A-rush (16:45-18:15 PM).

Methods	Entrance CFD Weekdays			Exit CFD Weekdays			Entrance CFD Weekends			Exit CFD Weekends			Average
	M-rush	Non-rush	A-rush	M-rush	Non-rush	A-rush	M-rush	Non-rush	A-rush	M-rush	Non-rush	A-rush	
OLS-AO	0.198	**0.347**	0.241	0.256	0.378	0.348	0.584	**0.519**	0.513	0.646	0.583	0.516	0.427
OLS-MR	0.193	0.357	**0.226**	0.247	0.382	**0.337**	0.608	0.558	0.551	0.637	0.592	0.563	0.437
OLS-DT	**0.190**	**0.341**	**0.226**	**0.233**	**0.369**	0.341	**0.578**	0.520	**0.507**	**0.631**	**0.580**	**0.508**	**0.418**
HA	0.209	0.369	0.248	0.269	0.401	0.348	0.634	0.598	0.592	0.689	0.639	0.642	0.470
LSM-RN-All	0.615	0.753	0.523	0.580	0.743	0.672	0.777	0.943	0.801	0.779	0.854	0.824	0.739
ONMF	0.440	0.633	0.420	0.578	0.675	0.679	0.660	0.779	0.791	0.817	0.775	0.715	0.664
SARIMA	0.225	0.424	0.266	0.311	0.436	0.446	0.730	0.630	0.631	0.818	0.669	0.676	0.522
GPR	0.801	0.491	0.592	0.868	0.503	0.644	0.738	0.592	0.568	0.793	0.656	0.629	0.656

Table 4.4: Overall results. We report the average errors among different methods between 6:00 AM and 10:00 PM. Best results are bold.

Methods	Entrance CFD Weekdays		Exit CFD Weekdays		Entrance CFD Weekends		Exit CFD Weekends		Average	
	MAE	MRE	MAE	MRE	MAE	MRE	MAE	MRE	MAE	MRE
OLS-AO	1.531	0.308	1.899	0.357	**1.912**	0.533	2.081	**0.590**	1.856	0.447
OLS-MR	1.542	0.311	1.887	0.356	1.944	0.557	2.129	0.601	1.875	0.457
OLS-DT	**1.520**	**0.289**	**1.855**	**0.350**	1.917	**0.528**	**2.077**	0.593	**1.842**	**0.440**
HA	1.652	0.334	1.969	0.391	2.176	0.617	2.635	0.667	2.108	0.502
LSM-RN-All	5.713	1.009	3.424	0.592	2.105	0.598	2.857	0.779	3.525	0.744
ONMF	4.122	0.763	2.687	0.502	2.119	0.603	3.417	0.883	3.086	0.688
SARIMA	1.939	0.395	2.304	0.463	2.270	0.659	2.671	0.724	2.296	0.560
GPR	4.283	0.955	4.928	1.001	2.333	0.671	2.590	0.662	3.533	0.822

Table 4.5: Comparisons on major stations. We report the average mean relative errors (MRE) of major stations. Best results are bold.

Methods	Entrance CFD Weekdays			Exit CFD Weekdays			Entrance CFD Weekends			Exit CFD Weekends			Average
	M-rush	Non-rush	A-rush	M-rush	Non-rush	A-rush	M-rush	Non-rush	A-rush	M-rush	Non-rush	A-rush	
OLS-AO	0.192	0.321	0.233	0.254	0.364	0.331	0.582	**0.510**	0.508	0.647	0.566	**0.512**	0.418
OLS-MR	0.187	0.344	0.219	**0.241**	0.389	0.321	0.601	0.549	0.526	0.633	0.572	0.545	0.427
OLS-DT	**0.183**	**0.315**	**0.210**	0.242	**0.356**	**0.317**	**0.574**	0.511	**0.501**	**0.619**	**0.559**	0.519	**0.408**
HA	0.211	0.386	0.244	0.271	0.398	0.352	0.627	0.611	0.599	0.671	0.643	0.651	0.472
LSM-RN-All	0.585	0.714	0.615	0.577	0.712	0.643	0.719	0.920	0.836	0.744	0.814	0.833	0.726
ONMF	0.454	0.612	0.391	0.558	0.665	0.699	0.661	0.740	0.698	0.887	0.741	0.736	0.654
SARIMA	0.221	0.408	0.273	0.331	0.452	0.419	0.721	0.647	0.632	0.800	0.675	0.632	0.518
GPR	0.765	0.501	0.541	0.797	0.530	0.634	0.719	0.607	0.566	0.777	0.634	0.610	0.644

Table 4.6: Comparisons with different time intervals. We report the average errors with different time interval between 6:00 AM and 10:00 PM. Best results are bold.

Methods	Time Interval – Half Hour				Time Interval – One Hour			
	Entrance CFD Weekdays		Exit CFD Weekdays		Entrance CFD Weekdays		Exit CFD Weekdays	
	MAE	MRE	MAE	MRE	MAE	MRE	MAE	MRE
OLS-AO	1.697	0.281	2.017	0.327	1.754	0.285	2.276	0.320
OLS-MR	1.703	0.285	1.939	0.311	1.807	0.292	2.019	0.315
OLS-DT	**1.654**	**0.274**	**1.901**	**0.305**	**1.750**	**0.279**	**1.985**	**0.308**
HA	1.988	0.327	2.334	0.381	2.133	0.333	2.556	0.371
LSM-RN-All	5.910	0.923	3.580	0.626	6.177	0.901	3.849	0.574
ONMF	4.567	0.722	3.076	0.477	4.257	0.734	3.111	0.463
SARIMA	2.198	0.390	2.255	0.461	2.459	0.382	2.354	0.396
GPR	4.372	0.913	4.883	0.898	4.473	0.901	5.013	0.921

Table 4.7: Transferability test on TaxiBJ dataset. We report the average mean relative errors (MRE) of city crowd flow prediction. Best results are bold.

Methods	Inflow weekdays	Outflow weekdays	Inflow weekends	Outflow weekends	Average
OLS-AO	0.164	0.165	0.172	0.173	0.169
OLS-MR	0.170	0.170	0.190	0.179	0.177
OLS-DT	**0.157**	**0.160**	**0.168**	**0.166**	**0.162**
HA	0.498	0.478	0.191	0.192	0.339
LSM-RN-All	0.276	0.298	0.246	0.310	0.282
ONMF	0.264	0.305	0.279	0.259	0.277
SARIMA	0.193	0.211	0.197	0.201	0.201
GPR	0.200	0.239	0.192	0.189	0.205

4.5.3 Results on the State-wide Train Network

We first assess the performances of our models across the state-wide train network. We consider two types of CFD flows (entrance and exit as discussed in section 4.2.2 and 4.2.3); comparisons between peak and non-peak times; and results between weekdays and weekends to evaluate our models comprehensively.

Table 4.3 reports the average *MRE* in different scenarios. We pick three noticeable time periods in this test, termed as M-rush (morning rush), Non-rush and A-rush (afternoon rush). In the weekday test, it is apparent that OLS-AO performs better during non-rush times, while OLS-MR performs better during rush times in both the entrance and the exit CFD prediction tasks. In the weekend test, OLS-AO achieves the better results because passenger flows are more stable throughout the day. The dual-track model which incorporates both OLS-AO and OLS-MR that performs almost the best through all experiments. Because OLS-AO and OLS-MR separately designed for the rush and non-rush hours, OLS-DT usually gets the closely results as the better one of the first two models. To keep image clear, we only draw top four methods in Figure 4.9 because other baselines perform far worse than the select methods. In addition, we remove the lines of OLS-DT because it overlaps closely with the best one of OLS-AO and OLS-MR that leads to a bad visualization in Figures. The tests drawn in Figure 4.9 demonstrate the conclusions above and the results of Table 4.3. For the regression methods, GPR and SARIMA, are both sensitive to the incomplete data, but SARIMA takes periodicity into consideration that gets the better results. They need to build models for every crowd flow that results in the huge re-training cost and the high complexity. Although, ONMF and LSM-RN-All can solve the network-wide prediction problem, they do not incorporate historic guidance that performance suffers. More comparisons are shown in Table 4.4.

Table 4.4 presents the average errors of all test methods for each timestamp between 6.00 am and 10.00 pm. OLS-AO provides the best performance on weekends compared with other baselines because sudden changes in flow are rare. However, on weekdays, OLS-AO and OLS-MR will outperform alternatively in different scenarios; in this case, the dual-track method achieve the best performance on weekdays.

In summary, the first two purpose-built models, OLS-AO and OLS-MR, have their own advantages for network-wide CFD predictions for complex metro system. OLS-DT achieves the best performances on weekdays and weekends.

4.5.4 Results for the Major Stations

This evaluation aims to remove any interference created by smaller stations to evaluate our methods' more comprehensively. In this case, we select the top 20 stations as the major stations based on throughput capacity. Table 4.5 shows the performances from three perspectives, including tests on entrance/exit CFD, weekdays and weekends, rush and non-rush hours. It is apparent that the experimental results lead to similar conclusions to our first test. The proposed three models outperform other baseline methods, OLS-AO and OLS-MR are able to adapt to different scenarios, and OLS-DT achieve the best overall results.

4.5.5 Results on Different Time Intervals

In the real-world crowd flow prediction application, an appropriate and practical time interval is around 15 minutes [12, 70, 78]. Authorities prefer a short-term prediction that can manage services more flexibly. We add this set of experiments to demonstrate the robustness of our models with different time intervals.

Table 4.6 represents the average MAE and MRE with different time intervals (half-hour and one-hour). In this weekday test, it is apparent that OLS-DT achieves better results than OLS-AO and OLS-MR, and these three proposed online models outperform other baselines. Even though that the spatial-temporal continuity will become weaker when the time interval increases, our models can learn periodicity knowledge from the historical guidance. This strategy leads to good performances when testing on the longer time intervals.

4.5.6 Visualization of Crowd Flow Distribution

Figure 4.10 gives an intuitive presentation of the CFD prediction in Sydney. The forecast time span is 3:45 PM – 4:00 PM on 7 Nov 2016. The first red marked value in the rectangle presents the predicted result of OLS-DT, while the value in the bracket is its ground truth. This visualization illustrates the effectiveness of our model. In this time period, people tend to go home (large residential areas, Hurstville and Burwood), and a small number of people are going to Central.

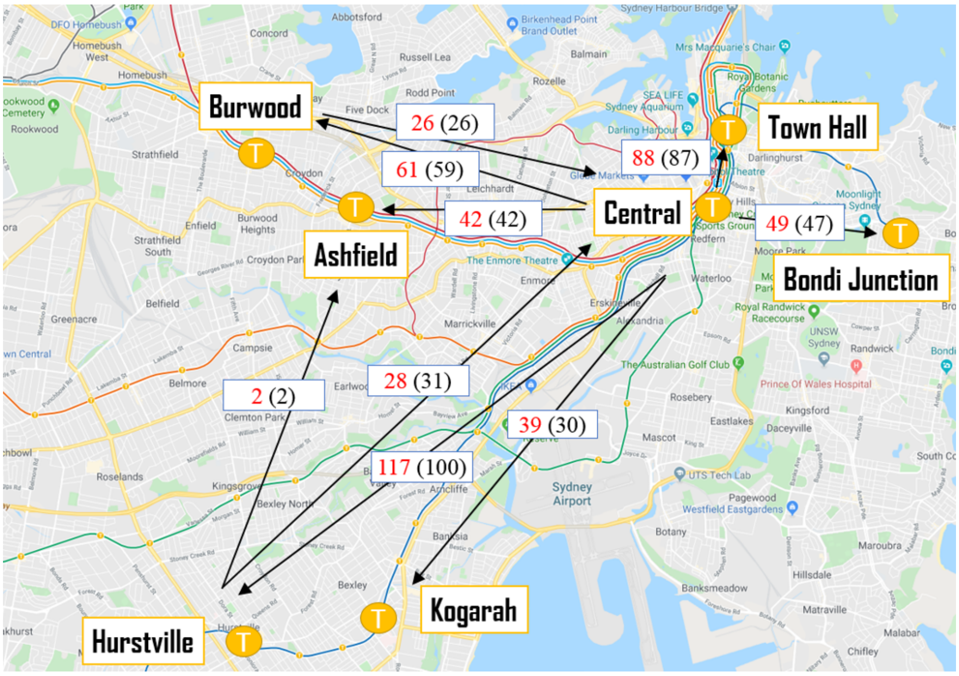


Figure 4.10: The visualization of CFD prediction.

4.5.7 Transferability Test on TaxiBJ Dataset

This set of experiments is designed to assess the transferability of our three models. OLS-AO, OLS-MR and OLS-DT are not only used for CFD prediction but suitable for other network-wide prediction problem, such as city-wide crowd flow forecasting. There are two types of flows of each city area, named as “Outflow” and “Inflow” where Outflow is the number of traffic crowds leaving an area to other regions in one timestamp, and Inflow presents the number of traffic crowds entering an area from other regions in one timestamp [97]. The time interval in this dataset is 30 minutes. Due to the fact that this problem does not suffer from the incomplete data challenge, we can remove the indication matrix P_t when conducting experiments.

Table 4.7 shows results of our methods together with some baselines. Compared with other methods, our three models can achieve the great accuracies. OLS-DT is performs the best on weekdays and weekends. In general, our methods perform better than other baselines that have the strong transferability to solve other crowd flow problem.

4.5.8 Ablation Study

In this section, we analyses the contribution of each component of the final optimization function. We only report the average mean relative errors on entrance CFD predictions

Table 4.8: Ablation studies on models. We report the average mean relative errors (MRE) of entrance CFD prediction on the entire trains network. Best results are bold.

Online	K_t	h	OLS-AO		OLS-MR	
			rush time	non-rush	rush time	non-rush
✓			0.430	0.633	0.425	0.701
✓	✓		0.408	0.611	0.375	0.667
✓		✓	0.236	0.361	0.217	0.364
✓	✓	✓	**0.220**	**0.347**	**0.210**	**0.357**

on weekdays because other parts of experiments obtain similar conclusions.

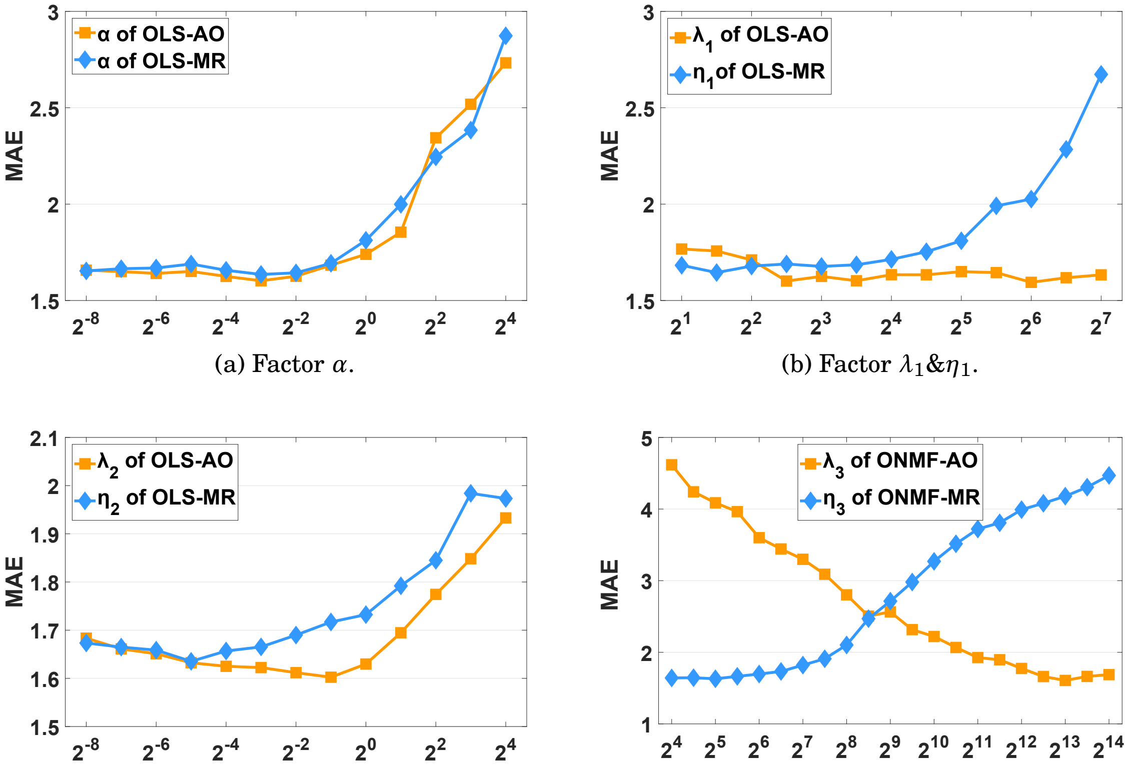
Table 4.8 illustrates that how the side information affect performance. The term "Online" means the online strategy with transition learning process, OLS-AO uses the average optimization and OLS-MR learns the most recent trend. K_t illustrates whether the model involve the inherent and external influences, e.g., station similarities and weather. h indicates the historical guidance is used or not.

As the tests shown in Table 4.8, we clearly see that two models perform bad when only considering the online strategy, that is also the reason why transitional ONMF model and LSM-RN-All cannot achieve good results. The other combinations show that the side information make great improvements of two models, especially for the historical guidance. This is mainly caused by the daily periodicity of crowd flows, which means CFDs are usually very stable and have a strong daily periodic property. The results of combination of "Online" and "h" represent the performances of our previous work in [22], which are slightly worse than the extended models. The inherent and external influences give a help for a better performance because it have drawn useful information from station correlations and the weather conditions.

4.5.9 The Sensitivity of Parameters

Table 4.2 presents the best choice of each parameter. To evaluate how they affect the performance of our methods, we varied them in a wide range in this test. In this section, we have only presented the outcomes for the entrance CFD on weekdays due to the page limitation.

Figure 4.11(a) shows the different performances with a varying setting for α. This factor controls the weights of weather effect. As can be shown in the results, $\alpha = 2^{-3}$ is



(a) Factor α.

(b) Factor λ_1&η_1.

(c) Factor λ_2&η_2.

(d) Factor λ_3&η_3.

Figure 4.11: Effect of parameters.

the best choice for both OLS-AO and OLS-MR. Beside, α is not sensitive in the range of 2^{-8} and 2^{-1}.

Figure 4.11(b) represents various results by varying λ_1 and η_1. These two parameters connect to our online process that control the strength of the current flow trends to use. λ_1=2^6 and η_1=$2^{1.5}$ achieve the best results for OLS-AO and OLS-MR, respectively.

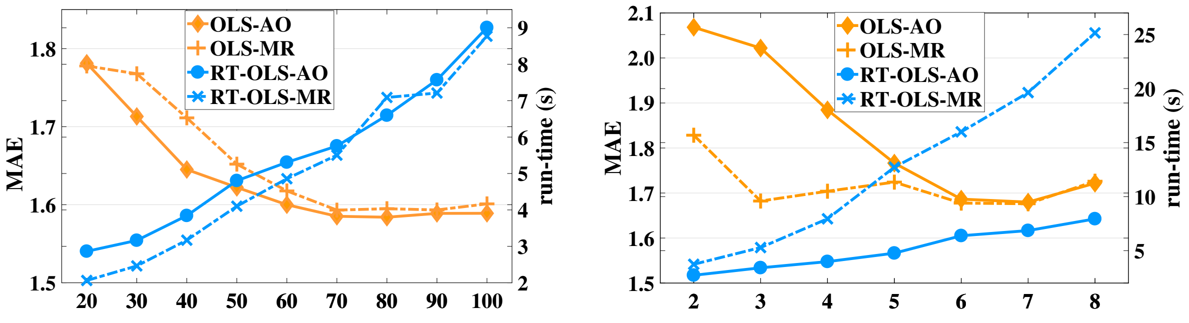
Figure 4.11(c) plots the effects with various λ_2 and η_2. The strength of guidance from inherent and external information are determined by these two parameters. λ_2=2^{-1} and η_2=2^{-5} yield the best performances for OLS-AO and OLS-MR, respectively.

Figure 4.11(d) illustrates the impacts of varying λ_3 and η_3 which effect the power of historic guidance. Generally, OLS-AO requires more historical information than OLS-MR to perform better. The best values of λ_3 and η_3 were 2^{13} and 2^5, respectively.

In a summary, the parameters used in our methods are benefit to the improvement of our models. Furthermore, they are not sensitively in a part of searching range which means our model are stable and easy fine-tuning.

Table 4.9: Scalability test. OLS-AO/OLS-MR completed each prediction step in a reasonable time span (about 5.5 seconds) with the highest accuracy.

	OLS-AO/MR	LSM	ONMF	SARIMA	GPR
re-train(s)	-	-	-	156.55	399.82
pred.(s)	5.53	3.69	4.96	0.68	0.36
train+pred.(s)	5.53	3.69	4.96	157.23	400.18



(a) Factor k.

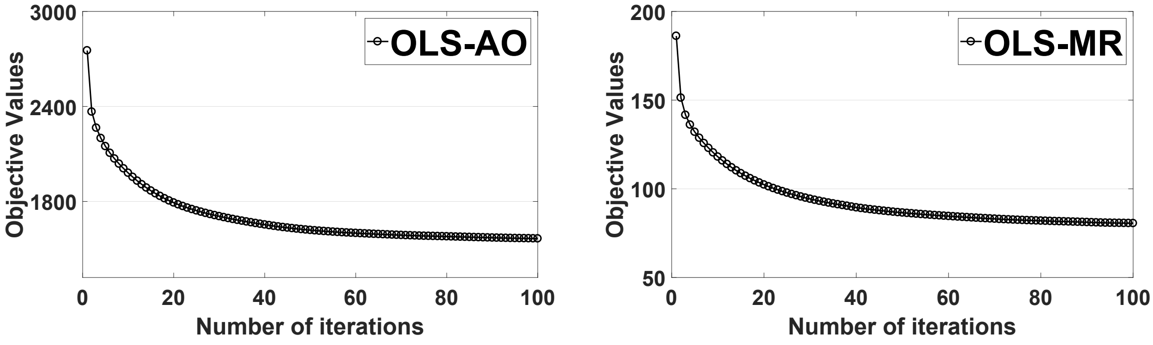
(b) Factor T.

Figure 4.12: Comparisons between running time and various k, T

4.5.10 Scalability

The scalability test is used to assess the efficiency of various approaches. We first to present the effects of window size T and the number of latent dimension k. Figure 4.12 (a) shows the various performances (left axis) and time costs (right axis) with various k. Generally, both OLS-AO and OLS-MR achieve the better performances but longer computational cost with k increased. Therefore, we set k to 70 for a good balance between computational cost and the prediction results. Figure 4.12 (b) shows the different prediction errors (left axis) and computational cost (right axis) with a varying setting for T. OLS-AO gets a minimum MAE when T=6, while OLS-MR performs good when $T \geq 3$. Hence, due to the computational cost will grow rapidly with T increases, we suggest setting T to 3 in our task.

Table 4.9 illustrates the results of the forecasting times and training process for one prediction step on the state-wide metro network based on the above recommended T and k. The regression methods, SARIMA and GPR, required an enormous amount of re-training time, which is difficult to implement in a real-time system. LSM and ONMF were faster at prediction than OLS-AO, OLS-MR but show a lower accuracy. The scalability test demonstrates that our models, OLS-AO and OLS-MR, are able to be



(a) OLS-AO

(b) OLS-MR

Figure 4.13: Convergence rate.

applied in the real-time network-wide problems, which cost approximately 5.5 seconds for each prediction step.

Convergence. As shown in Figures 4.13 (a) and (b), our models OLS-AO and OLS-MR can efficiently converge into a local optimization with a small number of iterations.

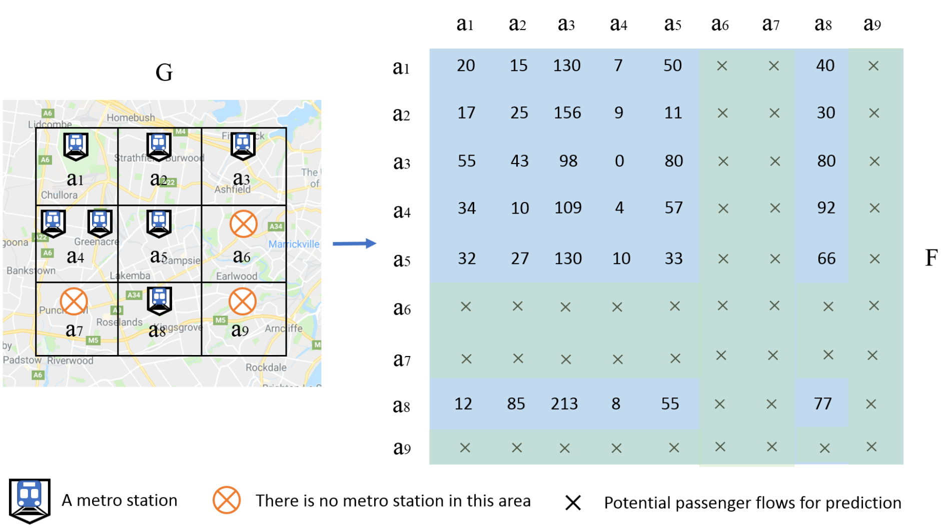
4.6 Conclusions

In this chapter, we proposed three spatiotemporal models to effectively address the network-wide CFD prediction problem based on the online latent space (OLS) strategy. The first model named OLS-AO combines the stable flow trends with the side information guidance by using an average optimization strategy that adapts to stable crowd flows. The second model, OLS-MR, learning from the most recent trends, is able to handle the dramatical changes of crowd flows. To enhance the models' applicability of handing real-world situations, our last dual-track model, OLS-DT, utilizes the benefits of both OLS-AO and OLS-MR to achieve the best performance in a variety of challenging traffic situations. All the empirical results demonstrate that the proposed models outperform all the other state-of-the-art methods.

CHAPTER 5

POTENTIAL PASSENGER FLOW PREDICTION

Recently, practical applications for passenger flow prediction have brought many benefits to urban transportation development. With the development of urbanization, a real-world demand from transportation managers is to construct a new metro station in one city area that never planned before. Authorities are interested in the picture of the future volume of commuters before constructing a new station, and estimate how would it affect other areas. In this chapter, this specific problem is termed as potential passenger flow (PPF) prediction, which is a novel and important study connected with urban computing and intelligent transportation systems. For example, an accurate PPF predictor can provide invaluable knowledge to designers, such as the advice of station scales and influences on other areas, etc. To address this problem, we propose a multi-view localized correlation learning method. The core idea of our strategy is to learn the passenger flow correlations between the target areas and their localized areas with adaptive-weight. To improve the prediction accuracy, other domain knowledge is involved via a multi-view learning process. We conduct intensive experiments to evaluate the effectiveness of our method with real-world official transportation datasets. The results demonstrate that our method can achieve excellent performance compared with other available baselines. Besides, our method can provide an effective solution to the cold-start problem in the recommender system as well, which proved by its outperformed experimental results.


G
a1
a2
a3
a4
a5
a6
a7
a8
a9
F
A metro station
There is no metro station in this area
× Potential passenger flows for prediction

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
a_1	20	15	130	7	50	×	×	40	×
a_2	17	25	156	9	11	×	×	30	×
a_3	55	43	98	0	80	×	×	80	×
a_4	34	10	109	4	57	×	×	92	×
a_5	32	27	130	10	33	×	×	66	×
a_6	×	×	×	×	×	×	×	×	×
a_7	×	×	×	×	×	×	×	×	×
a_8	12	85	213	8	55	×	×	77	×
a_9	×	×	×	×	×	×	×	×	×

Figure 5.1: The example of PPF prediction problem. We aim to forecast the passenger flows of target areas (e.g., a_6, a_7, a_9) across the entire city network.

5.1 Introduction

With the growth of intelligent transportation systems, passenger flow prediction models concentrate on discovering the volume of crowds and mobility patterns that best serve people's daily life [58, 98]. Recent advances in passenger flow prediction are focusing mainly on next time interval flow conditions with time evolves [22, 68]. If a brand-new metro station is inserted into the original metro network, existing predictors have to collect a large amount of latest transactional data to ensure normal operation. However, a real-world requirement from transportation authorities is that they want to obtain the potential passenger flows (PPF) of a planned city area in advance (i.e., before constructing a station in this area). It is significant for the urban traffic development and transportation management, as it can provide insights for the site selection of stations and analysis of passenger movement patterns, as well as give the potential crowd warning.

In the PPF prediction task, concentrating solely on the entrance and exit potential flows does not provide adequate information, authorities also desperately want to master the distribution of predicted PPF, i.e., forecast the number of potential passengers moving to different destinations. It is utmost important to find how will the new station affect other areas. For instance, Figure 5.1 illustrates an example of the PPF prediction problem. A city region is partitioned into nine areas[1], six of them have metro stations (termed as known areas), and three have not constructed yet (termed as target areas).

[1] We use grids for clear and simple illustration, the real partition standard is explained in the section of data description.

The right part of Figure 5.1 presents an origin-destination (OD) matrix (each row point is the origin area and column points are destinations), e.g., $F(a_1, a_3) = 130$ indicates that there are 130 passengers departure from a_1 and are going to the a_3. PPF task aims to make an accurate prediction for the target areas in one period (e.g., rush hours) that completes the crowd flows between them and known areas.

Traditional passenger flow mining usually deals with data from a single view. Recently, there exists a diversity of datasets from different sources in various domains with multiple views [102]. The multi-view learning algorithm is widely recognized as an effective way of solving the cross-domain problem, that features from different views can be served for the target domain learning process [17, 66, 87]. [88] proposed a matrix co-factorization based method (MVL-IV) to embed different views into a shared subspace, such that the incomplete views can be estimated by the information on observed views. To connect multiple views, MVL-IV assumed that different views have distinct ‚Äòfeature‚Äô matrices (i.e., $\{H_i\}_{i=1}^m$), but correspond to the same coefficient matrix (i.e., W). The tensor-based methods, such as [29], [30] [73] were proposed to address the cross-domain recommendation problem. They devised a cross-domain triadic factorization model to learn the triadic factors for user, item and domain, where the item dimensionality varies with domains. The above approaches and other similar methods [65, 86] cannot address our PPF prediction problem directly because they are not formulated for the passenger flow prediction task. However, since they can handle the cold-start problem by utilizing the cross-domain knowledge, an illuminative clue is educed. In conclusion, none of relevant studies can solve the PPF prediction problem directly. Accordingly, this chapter aims to design a reliable approach for PPF prediction with cross-domain knowledge involved.

To date, limited studies considered the OD passenger flow prediction problem [22, 81], and to the best of our knowledge, none of existing techniques can forecast PPF across the entire city. It is a novel problem and a real urban developing demand that faces several major challenges: (1) Considering the number of passenger flows and their final destinations simultaneously. (2) Analogously to the cold-start problem in the recommender system [33], it is hard to infer the preference of a new user from the known data. In our problem, a new station in the target area can be similarly regarded as a new user. (3) Since the PPF is a spatial-temporal mining problem, spatial and temporal information should be taken into account appropriately.

To resolve this novel and significant problem, in this chapter, we devise a multi-view localized correlation learning model for the PPF prediction (MLC-PPF for short). To leverage the spatial information, we first construct a localized similarity matrix which associates with the real geographical neighbors and regional properties (e.g., business or residential regions). The intuition behind this strategy is from the First Law of Geography [75], i.e. *"Everything is related to everything else, but near things are more related than distant things"*. Second, a novel weighted correlation learning strategy is proposed. At last, to improve the prediction accuracy and well handle the cold-start challenge, we draw the side information from urban statistical data, where each area has a multi-view features to guide the learning process. In summary, our main contributions are shown as follows:

- We formulate the PPF prediction problem and provide the first attempt on forecasting passenger flows for urban transportation development.
- We propose a multi-view localized correlation learning method to provide a solution for the PPF prediction that can learn localized correlations via a multi-view learning process.
- We show that our method can be transferred to the classic cold-start problem in the recommender system. It achieves a superior result that gives a new perspective for relevant tasks.
- We conduct extensive prediction experiments on a large real-world transactional dataset and show that our model outperforms other available algorithms.

5.2 Problem Statement

Focusing on the PPF prediction problem, every origin-destination among areas needs to be recorded. We formulate the OD passenger flow network as a fully connected graph $G = (A, E)$, where A is a set of vertexes and E is the set of edges. $a_i \subseteq V$ records the i-th origin or destination area, and an edge $e(v_i, v_j)$ denotes an origin-destination flow from area a_i to a_j. The value of each edge $e(a_i, a_j)$ is associated with the observed flow $f(a_i, a_j)$, i.e., $f(a_i, a_j)$ is the total number of passengers that departure from i^{th} area and are going to the j^{th} area. Then, G can be represented by PPF matrix $F = (f_{ij})$, where $f_{ij} = f(a_i, a_j)$. The example of G and F is shown in Figure 5.1. $f_{31} = 55$ means that 55 passengers leave from a_1 and theirs' destination is a_2.

In the real-world scenario, one area may have several stations. In this case, we calculate the passenger flows of these stations together to present the total flows of the

Table 5.1: Symbol description.

Symbols	Descriptions
G; a	crow flow network; the area
F_d	PPF matrix in the d-th day
F^k	the localized flow matrix of F
C	the localized correlation matrix
H	an indicator matrix for the k-nearest neighbors
W	the adaptive weight matrix
X_v	the v-th view of statistical data
Y	an indicator matrix for observed data
λ	regularization parameter

area. We consider three specific and useful time periods to predict PPF, which will help the authorities to do a better temporal analysis of transportation development. The three periods are morning rush hour (7.00 AM - 9.00 AM), afternoon rush hour (5.00 PM - 7.00 PM), and non-rush hour (2.00 PM - 4.00 PM).

Furthermore, traffic periodicity is a very important factor for relevant studies. Crowd flows also represent the stable and daily periodic properties, especially on weekdays. To extract the temporal information and make a more general prediction, we consider a series of previous daily PPF matrices (F_1,F_2,$\cdots$,F_D) in the same time period to predict the PPF matrix of target areas ($\hat{F_D}$) for the day D. Note that, the prediction is not limited in the D-th day, the target can be changed easily based on the real requirement.

To best simulate the crowd flow changes when picking the target areas, in this chapter, we tracked all trajectories of passengers, from origins to their destinations. For example, if area a_1 is selected as a target area, all the departure crowd flows from a_1 will add to its closest area (e.g., a_2) to best simulate the people's choice. In this way, the PPF is learned by the crowd flows under the assumption that the original passengers from a_1 will departure from its closest neighbor a_2.

5.3 The Proposed Method

In this section, we propose our PPF prediction model MCL-PPF. We will describe the strategy of localized correlation learning with adaptive-weight, and how to leverage the cross-domain multi-view information to improve our work. Figure 5.2 shows the flowchart of our model. For ease of presentation, the main symbols used in this chapter are summarized in Table 5.1.

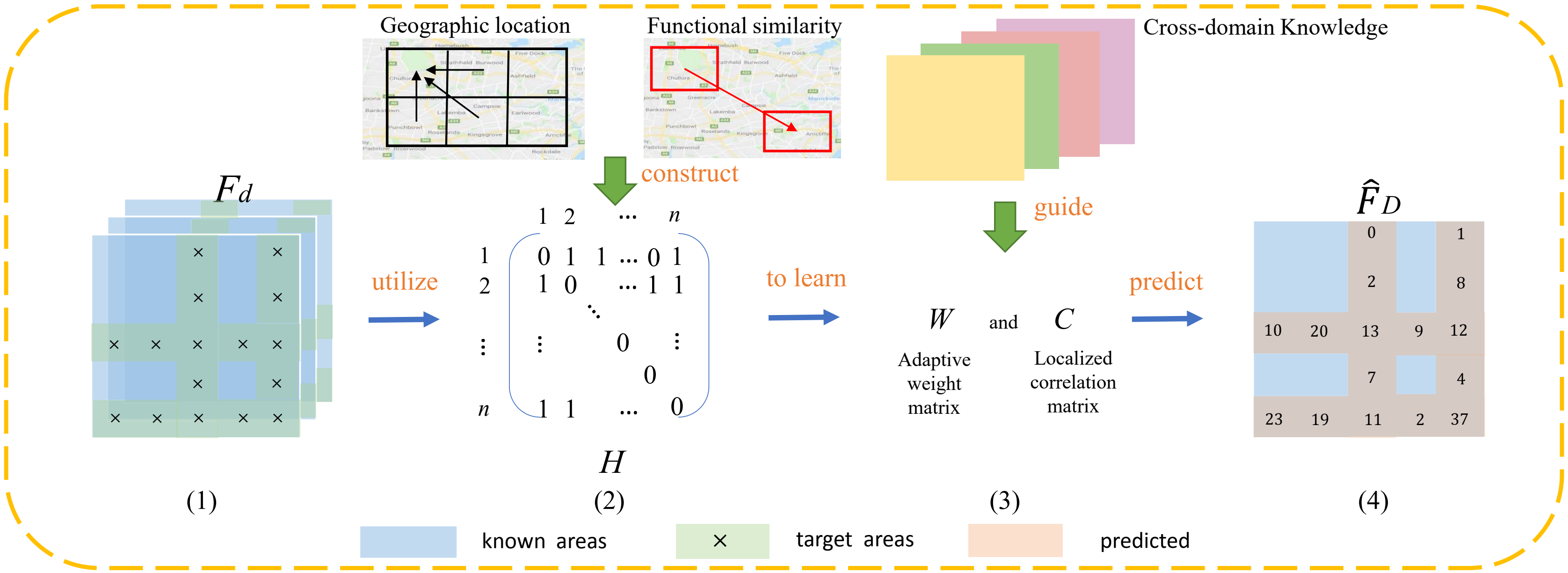


Figure 5.2: The flowchart of our proposed model. In the learning process, given a set of previous PPF matrices $\{F_d\}$, MLC-PPF learns the localized correlation matrix C and adaptive-weight W via a k-nearest indicator matrix H. The cross-domain knowledge is utilized to guide the updating of C. Then, the target prediction can be inferred by Algorithm 5.

5.3.1 Localized Correlation Learning

PPF prediction problem is a spatially related task that the more similar between two areas, the more correlations of passenger flow condition they have. Assuming that a city is partitioned into n areas, including known and target areas. $F \in \mathbb{R}^{n\times n}$ presents the PPF matrix, and $F^k \in \mathbb{R}^{n\times n}$ presents the localized flow matrix of F, where the i-th row of F^k is a combination of its k-nearest area passenger flows. In that way, we formulate the function to learn the localized correlation, which can be expressed as:

$$\begin{aligned} &\min_{C} \ \frac{1}{2}\sum_{d=1}^{D} \left\| F_d - F_d^k C \right\|_F^2 \\ &s.t. \ \ P_\Omega(F_d) = P_\Omega(F_d^k), \end{aligned} \tag{5.1}$$

where $C \in \mathbb{R}^{n\times n}$ is the localized correlation matrix that learns the transformation from F_d^k to F_d in each period of day d, D is the total number of days; $P_\Omega(\cdot)$ stands for the projection of all observed passenger flows from the known area set Ω; $\|\cdot\|_F$ is Frobenius norm of matrix.

Now, we will discuss how to build F_d^k of one period in a day. The physical distances among areas need to be considered first. Moreover, the development of a city gradually fosters different functional regions, such as business and educational areas, where the areas belonging to the same functional region will have strong connections with their properties [103].

Thus, the similarities among areas should take into account the above two standards. To this end, we build two distance metrics from the real geographic location and regional similarity. The distance metric between i-th and j-th areas is shown as follows:

$$s_{i,j} = 2 - \left(\frac{dist_{i,j}}{max(dist_{i,:})} + \frac{dist'_{i,j}}{max(dist'_{i,:})}\right), \tag{5.2}$$

where $dist_{i,j}$ is the geographic distance between i-th and j-th areas; and $dist'_{i,j}$ presents the Euclidean distance which is calculated by intrinsic features of areas (e.g., point of interest attributes).

After having gotten the $s_{i,j}$, the k neighbors of the i-th area can be picked. Then, we construct an indicator matrix H for the k-nearest neighbors of all areas where each row indicates the position of its k-nearest known areas. For example, in the stage 2 of Figure 5.2, the first row of H illustrates that a_2, a_3, and a_n are the k-nearest areas of

a_1 if k is setting to 3. Accordingly, for each day d, the localized flow matrix F_d^k can be represented as $F_d^k = (H \odot W)F_d$, where W is an adaptive-weight matrix that learns the different weights of k-nearest areas. To this stage, the localized correlation learning process is shown as Equation (5.3).

$$\begin{aligned} \min_{C,W} \ & \frac{1}{2}\sum_{d=1}^{D} \|F_d - (H \odot W)F_d C\|_F^2 \\ s.t. \ & P_\Omega(F_d) = P_\Omega(F_d^k), \end{aligned} \tag{5.3}$$

where $\odot$ is the entrywise product; The loss function aims to learn the localized correlations matrix C and weight matrix W simultaneously.

5.3.2 Improvement by Cross-domain Learning Process

As mentioned above, there are various functional regions of a city. Thanks to the urban statistical data, the passenger flow similarities among different areas can be reviewed from this cross-domain perspective. Based on the phenomenon that the similar functional regions have the similar passenger flows (e.g., the business regions have a large number of entrance flows during the morning rush hour, and people leave from residential areas in the same time span), we leverage such information to guide the localized correlation learning process.

The statistical data have multiple views to record the differences between areas. For example, the economy view reveals the economic features, such as the number of industries and employee statistics; and the population view consists of detailed population information. Let $\{X_1, X_2, \cdots, X_V\}$ denote the multi-fold views of statistical data, where $X_v \in \mathbb{R}^{n \times m_v}$, the row of X_v denotes the area and column denotes the feature. To improve the prediction performance, cross-domain knowledge is involved as guidance, which can be formulated as:

$$\begin{aligned} \min_{C,W} \ & \frac{1}{2}\sum_{d=1}^{D} \|F_d - (H \odot W)F_d C\|_F^2 + \frac{\lambda}{2}\sum_{v=1}^{V} \left\| X_v - C X_v^k \right\|_F^2 \\ s.t. \ & P_\Omega(F_d) = P_\Omega(F_d^k), \end{aligned} \tag{5.4}$$

where λ is the regularization parameter; $X_v^k = H X_v$ denotes the localized matrix of X_v.

After solving Equation (5.4), the learned matrices W and C can be used to make the prediction. The predicted PPF of target areas in D-th day is:

$$\hat{F_D} = (1 - Y) \odot ((H \odot W) F_D C), \tag{5.5}$$

where Y is an indicator matrix whose entry (i,j) is one if $F(i,j)$ is observed and zero otherwise.

To this stage, the OD passenger flows in the target areas are learned by the above processes, i.e., predict each row of target areas. However, the column of target areas revealing how much crowds arrived at these areas needs to be predicted with a slight modification. That is, replace F_d with $F_d^\top$ in Equation (5.4) to learn the localized correlation from the other side. It can be solved in a likewise manner. Thus we only presented the optimization strategy of Equation (5.4) due to the page limitation.

5.3.3 Learning and Prediction

Equation (5.4) is a complex non-convex problem. But the loss function associated with Equation (5.4) is convex regarding C with fixed W and vice verse. We can optimize them alternatively until convergence (e.g., alternating least squares (ALS)). A straightforward way to minimize the loss function is through the gradient method.

Considering C while W is fixed, Equation (5.4) can be rewritten as follows:

$$\begin{aligned} \mathscr{L} =& \frac{1}{2} \sum_{d=1}^{D} Tr((F_d - (H \odot W) F_d C)(F_d - (H \odot W) F_d C)^\top) \\ &+ \frac{\lambda}{2} \sum_{v=1}^{V} Tr((X_v - C X_v^k)(X_v - C X_v^k)^\top). \end{aligned} \tag{5.6}$$

Taking the derivative of $\mathscr{L}$ with respect to C, we can get gradient gC:

$$\begin{aligned} gC = & \sum_{d=1}^{D} ((H \odot W) F_d)^\top ((H \odot W) F_d C - F_d) \\ & + \lambda \sum_{v=1}^{V} (C X_v X_v^{k\top} - X_v X_v^{k\top}). \end{aligned} \tag{5.7}$$

Analogously, the derivative of $\mathscr{L}$ with respect to W is:

$$gW = \sum_{d=1}^{D} (H \odot W) F_d C C^\top F_d^\top - H \odot (F_d C^\top F_d^\top). \tag{5.8}$$

Let α and T be the step-size and number of iterations. In each stage, we adopt the following update rules:

$$C_{t+1} = C_t - \alpha \frac{gC_t}{\|gC_t\|_F}, \tag{5.9}$$

$$W_{t+1} = W_t - \alpha \frac{gW_t}{\|gW_t\|_F}, \tag{5.10}$$

$$F_{d(t+1)} = Y \odot F_{d(t)} + (1-Y) \odot ((H \odot W_t) F_{d(t)} C_t), \tag{5.11}$$

where $t = 1, 2, \cdots, T$.

We give the derivatives of $\mathscr{L}$ with respect to C and W. The objective function of $\mathscr{L}$ can be rewritten as follows:

$$\begin{aligned} \mathscr{L} = &-\frac{1}{2}\sum_{d=1}^{D} Tr((H \odot W)F_d C\hat{F_d}^\top) \\ &-\frac{1}{2}\sum_{d=1}^{D} Tr(\hat{F_d}^\top C^\top F_d^\top (H \odot W)^\top) \\ &+\frac{1}{2}\sum_{d=1}^{D} Tr((H \odot W)F_d C C^\top F_d^\top (H \odot W)^\top) \\ &-\frac{\lambda}{2}\sum_{v=1}^{V} Tr(CX_v^k X_v^\top) - \frac{\lambda}{2}\sum_{v=1}^{V} Tr(X_v X_v^{k\top} C^\top) \\ &+\frac{\lambda}{2}\sum_{v=1}^{V} Tr(CX_v X_v^{k\top} C^\top) + \mathscr{R}(\bar{C}), \end{aligned} \tag{5.12}$$

where $\mathscr{R}(\bar{C})$ indicates the components of $\mathscr{L}$ excluding C, which is a constant value when only considering the partial derivative of C. The further simplification of Equation (5.12) is:

$$\begin{aligned} \mathscr{L} = &-\sum_{d=1}^{D} Tr((H \odot W)F_d C\hat{F_d}^\top) \\ &+\frac{1}{2}\sum_{d=1}^{D} Tr((H \odot W)F_d C C^\top F_d^\top (H \odot W)^\top) \\ &-\sum_{v=1}^{V} Tr(CX_v^k X_v^\top) + \frac{\lambda}{2}\sum_{v=1}^{V} Tr(CX_v X_v^{k\top} C^\top) + \mathscr{R}(\bar{C}). \end{aligned} \tag{5.13}$$

Taking the derivative of $\mathscr{L}$ with respect to C, we can get:

$$\begin{aligned}
g(C) = & -\sum_{d=1}^{D} F_d^\top (H \odot W)^\top \hat{F_d} + \frac{1}{2}\sum_{d=1}^{D} F_d^\top (H \odot W)^\top (H \odot W) F_d C \\
& + \frac{1}{2}\sum_{d=1}^{D} F_d^\top (H \odot W)^\top (H \odot W) F_d C - \lambda \sum_{v=1}^{V} X_v X_v^{k\top} \\
& + \frac{\lambda}{2}\sum_{v=1}^{V} C X_v^k X_v^{k\top} + \frac{\lambda}{2}\sum_{v=1}^{V} C X_v^k X_v^{k\top} \\
= & -\sum_{d=1}^{D} F_d^\top (H \odot W)^\top \hat{F_d} + \sum_{d=1}^{D} F_d^\top (H \odot W)^\top (H \odot W) F_d C \\
& - \lambda \sum_{v=1}^{V} X_v X_v^{k\top} + \sum_{v=1}^{V} C X_v^k X_v^{k\top} \\
= & \sum_{d=1}^{D} ((H \odot W) F_d)^\top ((H \odot W) F_d C - \hat{F_d}) \\
& + \lambda \sum_{v=1}^{V} (C X_v X_v^{k\top} - X_v X_v^{k\top}).
\end{aligned} \tag{5.14}$$

Analogously, the objective function of $\mathscr{L}$ can be rewritten as:

$$\begin{aligned}
\mathscr{L} = & \sum_{d=1}^{D} Tr((H \odot W) F_d C \hat{F_d}^\top) \\
& + \frac{1}{2}\sum_{d=1}^{D} Tr((H \odot W) F_d C C^\top F_d^\top (H \odot W)^\top) \\
& + \mathscr{R}^{'}(\bar{W}),
\end{aligned} \tag{5.15}$$

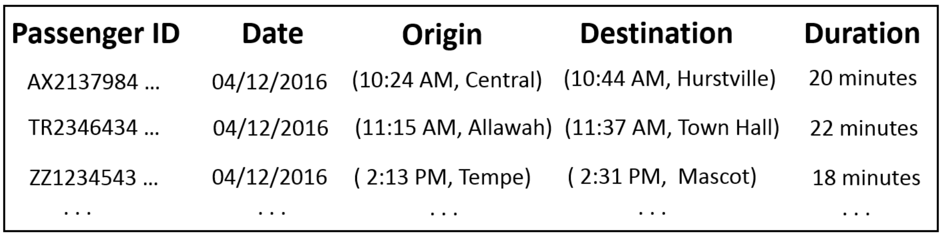
where $\mathscr{R}^{'}(\bar{W})$ indicates the components of $\mathscr{L}$ excluding W, which is a constant value when only considering the partial derivative of W. Then, the derivative of $\mathscr{L}$ with respect to W is:

$$\begin{aligned}
g(W) = & -\sum_{d=1}^{D} H \odot (\hat{F_d} C^\top F_d^\top) + \frac{1}{2}\sum_{d=1}^{D} (H \odot W) F_d C C^\top F_d^\top \\
& + \frac{1}{2}\sum_{d=1}^{D} (H \odot W) F_d C C^\top F_d^\top \\
= & \sum_{d=1}^{D} (H \odot W) F_d C C^\top F_d^\top - H \odot (\hat{F_d} C^\top F_d^\top).
\end{aligned} \tag{5.16}$$

Based on the above update equations, the iterative learning and prediction process for MLC-PPF are summarized in Algorithm 5.

Algorithm 5: MLC-PPF

```
Input: PPF matrices $[F_1,\cdots,X_D]$; Mutiple views of statistical data $[X_1,\cdots,X_V]$.
Output: Prediction $F_D$
Initialize $C$: $C \leftarrow (Y \odot (H \odot W)F_D))^{\dagger}(Y \odot F_D)$ by solving Equation (5.3), where † the pseudo-inverse of matrix;
Initialize $W$: $W \leftarrow S$, where $S$ is built by Equation (5.2)
Construct $H$ by the real geographic location and regional similarity.
for $t$ = 1 to $T$ do
    if $|\mathscr{L}_t - \mathscr{L}_{t+1}|$ / $\mathscr{L}_t \geq \varepsilon$ then
        update $C_{t+1}$ By Equation 5.9
        update $W_{t+1}$ By Equation 5.10
        update $F_{t+1}$ By Equation 5.11
    else
        Break
Return $\hat{F_D}$ By Equation 5.5.
```



Passenger ID	Date	Origin	Destination	Duration
AX2137984 ...	04/12/2016	(10:24 AM, Central)	(10:44 AM, Hurstville)	20 minutes
TR2346434 ...	04/12/2016	(11:15 AM, Allawah)	(11:37 AM, Town Hall)	22 minutes
ZZ1234543 ...	04/12/2016	(2:13 PM, Tempe)	(2:31 PM, Mascot)	18 minutes
...	...	...	...	...

Figure 5.3: Examples of transactional data.

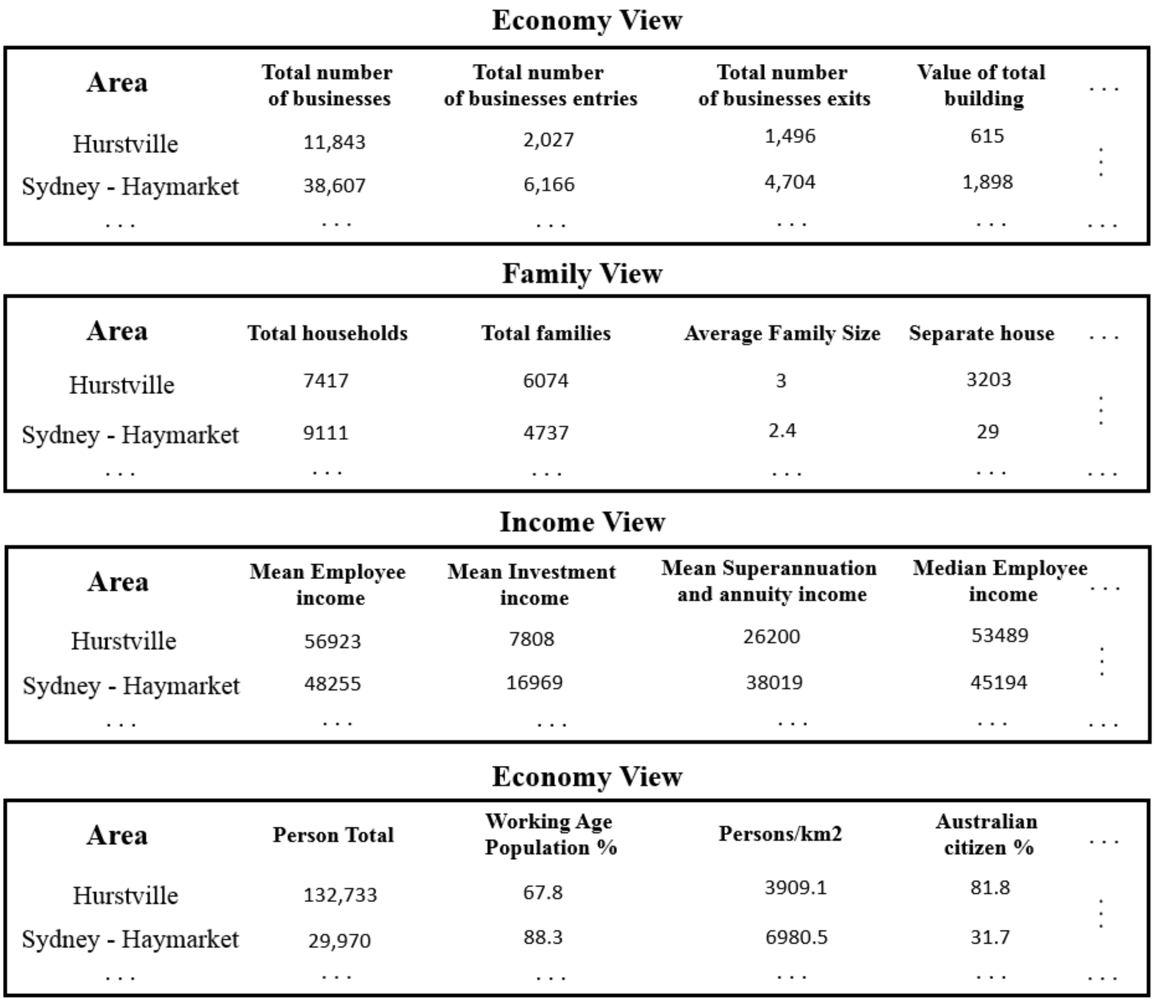
5.4 Experiments

In this section, we report on the experiments carried out on the real-world dataset, and show the effectiveness of our proposed method.

5.4.1 Data Description

- We describe the transactional dataset used in this chapter, which is a large-scale, real-world dataset provided by NSW Sydney Transport. After data cleaning[2], the dataset contains above 35 million transactional records covering 194 stations including the city train and ferry stations between 7 Nov 2016 and 11 Dec 2016. We pick the data between 7 Nov. 2016 and 20 Nov. 2016 as the training and validation sets (used to tune parameters); the remaining data are used as the test set.

[2] We removed the recording errors and UNKNOWN trips, etc.



Economy View

Area	Total number of businesses	Total number of businesses entries	Total number of businesses exits	Value of total building	. . .
Hurstville	11,843	2,027	1,496	615	⋮
Sydney - Haymarket	38,607	6,166	4,704	1,898	
. . .	. . .	. . .	. . .	. . .	. . .

Family View

Area	Total households	Total families	Average Family Size	Separate house	. . .
Hurstville	7417	6074	3	3203	⋮
Sydney - Haymarket	9111	4737	2.4	29	
. . .	. . .	. . .	. . .	. . .	. . .

Income View

Area	Mean Employee income	Mean Investment income	Mean Superannuation and annuity income	Median Employee income	. . .
Hurstville	56923	7808	26200	53489	⋮
Sydney - Haymarket	48255	16969	38019	45194	
. . .	. . .	. . .	. . .	. . .	. . .

Economy View

Area	Person Total	Working Age Population %	Persons/km2	Australian citizen %	. . .
Hurstville	132,733	67.8	3909.1	81.8	⋮
Sydney - Haymarket	29,970	88.3	6980.5	31.7	
. . .	. . .	. . .	. . .	. . .	. . .

Figure 5.4: Examples of ABS data.

Transactional database stores a large number of activity attributes of passengers. We only used the records that are related to our problem and fulfilled the confidentiality deed that ensures the privacy of each passenger. The useful records include: **(1) Passenger**: a unique hashed identification number; **(2) Origin**: the time and location where the passenger started a journey; **(3) Destination**: the time and location where the passenger ended a journey; **(4) Duration**: the number of seconds taken to complete a journey. An example of data is presented in Figure 5.3.

- The urban statistical data are collected from Australian Bureau of Statistics 2017 (ABS) with four views; those are Economy, Family, Income, and Population. In this chapter, we used the statistics of Sydney and the numbers of dimension of four

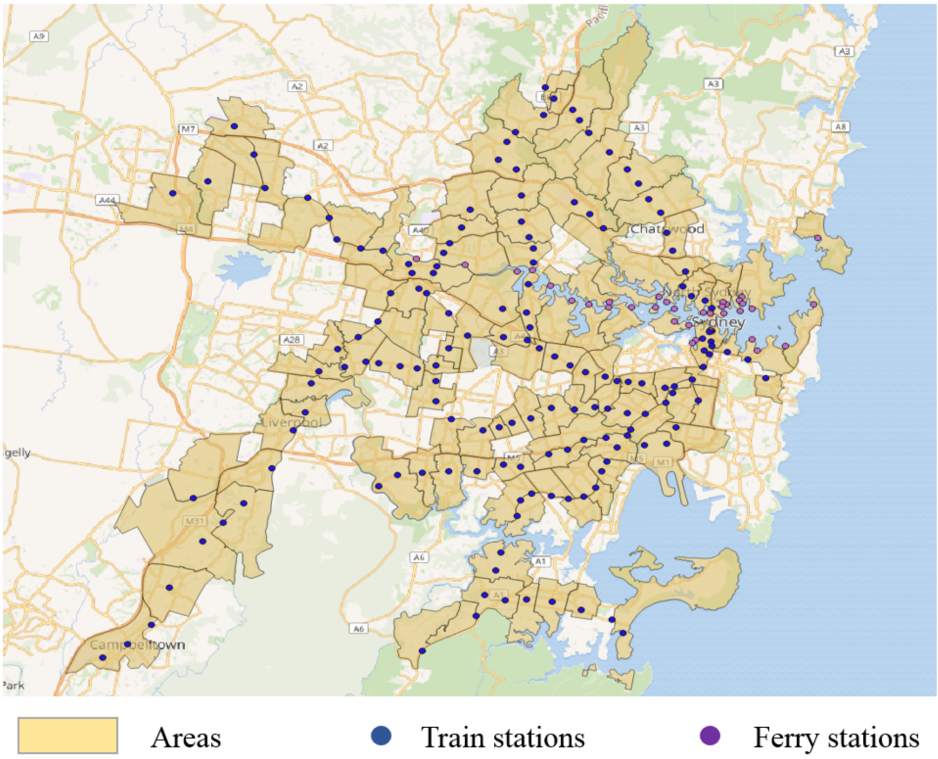


Figure 5.5: City partition and station mapping.

views are 43, 44, 50, 97, respectively. An example of data is presented in Figure 5.4. In our method, we normalized all data for the cross-domain guidance.

- All the transactional dataset across the transport network are mapped into 117 areas to build the flow matrices F_d, $d = 1,...,D$. The designation of areas is based on the Australian Statistical Geography Standard for the best practical value.

  Figure 5.5 illustrates the city partition and station mapping. The designation of areas based on Australian Statistical Geography Standard have been painted yellow. In this Figure, the mapping between stations and areas are given. The reason we only choose the areas existing stations is that the ground-truth can be obtained, and it is difficult to evaluate the blank area.

5.4.2 Methods and Metrics

We use the following five baselines which can learn the flow data by the cross-domain knowledge guidance. Among them, CDTF and WITF are two tensor-factorization-based (TF) methods that can solve the cold-start problem. For NMF, we concatenate the flow matrix with the statistical data. All parameters used in baselines and our method are picked by a grid search approach.

- **NMF**: Predict the PPF by the non-negative matrix factorization, which concatenates the flow matrix and the statistical data [35].
- **MVL-IV**: A state-of-the-art multi-view learning method based on the matrix co-factorization, it learns the same coefficient matrix to connect multiple views [88]. In this method, we set the flow matrix as one of the views, other views are from the ABS dataset.
- **LS-KNN**: Latent similarity k-nearest neighbors. After calculating the latent similarities among areas by Equation (5.2), we pick k-nearest neighbors of the target areas, and use average crowd flows of these neighbors as an estimate (k=4).
- **CDTF**: A state-of-the-art TF method to learn the cross-domain knowledge [29].
- **WITF**: A weighted irregular TF method which is similar as the CDTF [30]. For CDTF and WIFT, we leverage the passenger flow and ABS data to construct the tensor.

Metrics. We used the two most widely used evaluation metric to measure the PPF prediction quality. They are *Mean Absolute Error* (MAE) and *Normalized Root Mean Square Error* (NRMSE).

$$MAE = \frac{\sum_{i,j=1}^{M} |f_{ij} - \hat{f}_{ij}|}{M},$$

$$NRMSE = \frac{100\%}{nval} \sqrt{\frac{1}{M} \sum_{i,j=1}^{M} (f_{ij} - \hat{f}_{ij})^2},$$

where $\hat{f}_{ij}$ is a forecasting passenger flow from i-th area to j-th; and f_{ij} is the ground truth; M is the number of predictions; $nval = max(f_{ij}) - min(f_{ij})$.

Table 5.2: Comparisons with different time periods. We report the average mean absolute errors (MAE) and normalized root mean square error (NRMSE) among various methods. The target areas occupied 20% of the total set. Best results are bold.

Methods	Morning Rush Hour		Afternoon Rush Hour		Non-rush Hour		Average	
	MAE	**NRMSE**	**MAE**	**NRMSE**	**MAE**	**NRMSE**	**MAE**	**NRMSE**
NMF	124.50	30.78%	117.92	37.13%	89.11	28.44%	110.51	32.12%
MVL-IV	108.31	29.50%	101.55	29.78%	92.05	27.54%	100.64	28.94%
CDTF	75.15	22.43%	84.02	25.93%	67.78	19.37%	75.65	22.58%
WITF	69.30	18.73%	72.06	19.45%	62.57	17.26%	67.98	18.48%
LS-KNN	19.89	5.42%	20.20	7.67%	23.51	7.94%	21.20	7.01%
MLC-PPF	**9.84**	**2.30%**	**11.47**	**3.12%**	**8.22**	**1.21%**	**9.84**	**2.21%**

Table 5.3: Comparisons with different removing ratios. We report MAE and NRMSE through all test data.

Methods	5%		10%		15%		25%	
	MAE	**NRMSE**	**MAE**	**NRMSE**	**MAE**	**NRMSE**	**MAE**	**NRMSE**
NMF	88.11	20.70%	90.23	21.66%	107.44	26.52%	128.70	30.11%
MVL-IV	92.75	20.36%	90.40	20.75%	99.01	23.79%	101.93	32.40%
CDTF	59.25	18.90%	61.25	19.26%	68.07	22.00%	79.06	26.35%
WITF	60.41	18.23%	60.72	19.77%	61.18	19.23%	71.07	20.73%
LS-KNN	13.69	4.72%	16.45	5.01%	18.51	5.44%	23.99	9.25%
MLC-PPF	**8.64**	**1.37%**	**8.80**	**1.20%**	**9.73**	**1.90%**	**11.07**	**2.34%**

5.4.3 Comparisons on Different Time Periods

The first set of experiments is designed to assess the performance on different time periods. We randomly removed 20% areas as the target set, and the remaining 80% areas as the known set. The learning step α is fitted to 10^{-2} and there are only two hyper-parameters used in our method, where k and λ are chosen from {1,2,3,...,10} and $\{10^{-5}, 10^{-4}, ..., 10^{5}\}$ respectively. We repeat the experiment 20 times with random initialization and report the average results.

Experimental results are presented in Table 5.2. Compared with other approaches, our method achieved the best prediction accuracies on both three time periods. None of the multi-view and cross-domain methods work well because it is hard to capture the relationships between statistical data and the passenger flows. The approach LS-KNN performs better than other baselines, which illustrates that the PPF prediction problem has a strong spatial correlation property. In summary, the proposed method is a well-designed model for PPF prediction, which outperforms the other available baselines because it considers the localized correlations and the cross-domain knowledge simultaneously.

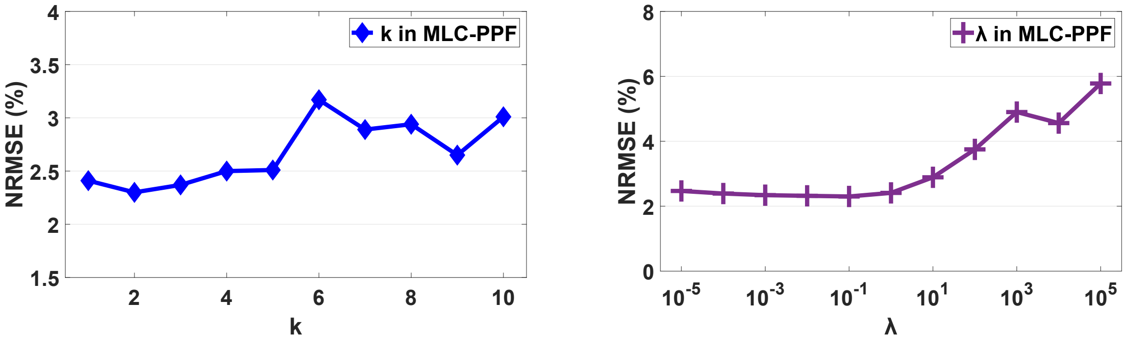
5.4.4 Comparisons on Various Missing Ratios

In this experiment, we evaluate how performance will change with varied number of target areas. We randomly pick 5%, 10%, 15%, 25% areas as the target areas, and run 20 times to report the average errors. The test period is in the morning rush hours. The performances of different methods are summarized in Table 5.3.

It is apparent that the experimental results lead to similar conclusions to the first comparison. Our model, MLC-PPF, significantly outperforms all other comparative methods over all testing sets. The performances of MLC-PPF in 5% dataset are very close to that of in 10%, which illustrates that the 90% remaining area set can learn a satisfied localized correlation and make accurate PPF predictions. In the real-world application, the proportion of target areas is usually small since only a few areas are suitable for constructing a new station.

5.4.5 Ablation Study

In this section, we analyze the contribution of critical components of the final optimization function. We report the average mean relative error and normalized root mean square



(a) Factor k.

(b) Factor λ.

Figure 5.6: Effect of parameters.

Table 5.4: Ablation Studies on our method. We report how the adaptive matrix W and ABS guidance affect the performance. The average MAE and NRMSE conducted on the morning rush period are shown below.

W	ABS	10%		20%	
		MAE	NRMSE	MAE	NRMSE
✓		9.72	2.07%	10.11	2.51%
	✓	10.34	1.98%	12.62	2.43%
✓	✓	**8.80**	**1.20%**	**9.84**	**2.30%**

error here on the morning rush period with two removing ratios since the similar conclusions can be gotten on other time periods.

Table 5.4 illustrates how the adaptive-weight matrix W and cross-domain knowledge from ABS data affect the performance of our model. The field "W" means whether we learn the adaptive weight of k-nearest neighborhoods, and field of "ABS" denotes that the ABS data is involved or not. As the tests shown in Table 4.8, it is apparent that our method performs worse when only considering the adaptive-weight matrix or ABS guidance. On the one hand, our model achieves a better MAE when the adaptive-weight W worked solely, but performs worse based on $NRMSE$. On the other hand, the ABS guidance is useful to avoid the abnormal prediction because it yields better results on $NRMSE$.

5.4.6 Parameter Analysis

In this section, we analyze the effects of two hyper-parameters used in this chapter, where k is the number of nearest neighborhoods, and the regularization factor λ controls the strength of guidance from ABS data.

Figure 5.6(a) shows the different performances with a varying setting for k. For each area, the correlation matrix C only learns the transform from these neighborhoods. As can be shown in the results, $k = 2$ is the best choice for our method, and k is not sensitive in the range of 1 and 5.

Figure 5.6(b) represents various results by varying λ. λ=2^{-1} achieves the best results for our method, and the performances are stable when choosing between [10^{-5}, 10^{0}].

In a summary, the parameters used in this chapter are benefit to the improvement of our models. MLC-PPF is stable because it is insensitive to parameters.

5.4.7 Case Study

We display a PPF prediction result of one target area in this section. In this case, the area “Homebush” is treated as the target area. For better visualization, we only remain the areas where the number of arrived passengers is greater than 5.

As shown in Figure 5.7, our model yields a great prediction result compared with the ground-truth, especially in some main areas of Sydney, such as the central area “Sydney-Haymarket”, “Burwood-Croydon”, “North Sydney-Lavender Bay” and “Parramatta-Rosehill”. The case study demonstrates the effectiveness of our method for the PPF prediction.

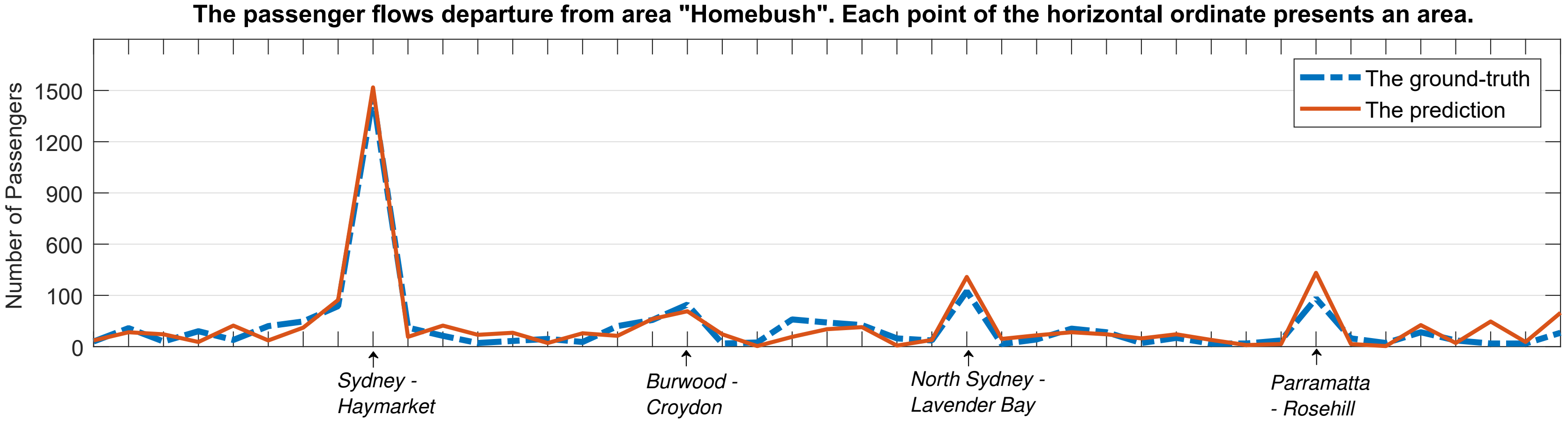


Figure 5.7: The case study. This figure shows the passenger flow prediction that departure from “Homebush” to other areas. To keep figure clear, we only draw our method and the ground-truth because other baselines perform far worse than the MLC-PPF.

Table 5.5: Transfer to the cold-start problem. We report the MAE of all test methods.

Target Domain	CMF	CDTF	WITF	MLC-PPF
Book	0.834	0.755	0.740	**0.396**
Music	0.847	0.779	0.716	**0.582**

5.4.8 Transfer to the Cold-start Problem

As we have emphasized, our strategy can provide a new perspective to address the classic cold-start problem in the recommender system. This set of experiments is designed to assess the transferability of our model.

We choose a very famous dataset from Amazon to do the evaluation, in which the dataset contains 1,555,170 users and 1-5 scaled ratings over 548,552 different products covering four domains: books, music CDs, DVDs and videos [29]. We randomly remove the 20% users from target domains to simulate the cold-start situation. Three baselines are used in this comparison. CMF is an effective method based on the collective matrix factorization which couples rating matrices for all domains on the User dimension [66]. CDTF and WITF are two tensor-factorization-based cross-domain recommendation methods, they devise a strategy to transform original data into a cubical tensor that can better capture the interactions between user factors and item factors [29, 30]. In this experiment, we leverage the information excluding target domain to build the k-nearest indicator matrix H.

Table 5.5 shows the results of our methods together with some state-of-the-art approaches. MLC-PPF can achieve the greatest accuracies for the target domains, which illustrates that our method is able to solve the unacquainted world phenomenon and give inspiration for relevant tasks. Despite the effectiveness of our methods, we should admit that there is a limitation of MLC-PPF. MLC-PPF only can make the prediction when its k-neighbors have ratings. However, based on the test results, the predicted ratings are reliable and able to make the recommendation.

5.5 Conclusion

In this chapter, we proposed an effective method for the potential passenger flow prediction, which is a novel study that brings benefits to the urban transportation development. To address this spatio-temporal problem, we design a multi-view localized correlation learning model (MLC-PPF) for the PPF prediction. The k-nearest indicator matrix H is constructed by the real geographical neighbors and regional properties. MLC-PPF can learn the correlations between each known area and its k-nearest neighbors with the cross-domain knowledge guidance. We evaluate the performance of our method with a set of well-designed experiments. All empirical results not only demonstrate that the proposed model outperforms all the other methods in the PPF prediction task, but also represent the capability of tackling the cold-start problem in recommender system.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

This thesis presents several methods to address three main predictive tasks in the network-wide crow flow prediction for the intelligent transportation system. We first address the spatio-temporal missing data problem; and second, to improve the quality of transportation services, we proposed three spatiotemporal models to effectively address the network-wide crowd flow distribution (CFD) prediction problem based on the online latent space (OLS) strategy. Last, we provide the first attempt on the potential passenger flow (PPF) prediction problem.

Chapter 3 proposes a spatial missing data imputation method for multi-view urban statistical data, called SMVNMF. To address the multi-view problem, an improved spatial multi-kernel method is designed to guide the imputation process based on the NMF strategy. Moreover, the spatial correlations among different regions are involved in our method from two perspectives. Firstly, the latent similarities are discovered by S-MKKN and S-KKM based on the idea of finding functional regions, and secondly, KNN is used for capturing the information of real geographical positions. We conduct intensive experiments on six real-world datasets to compare the performance of our model and other state-of-the-art approaches. The results not only show that our approach outperforms all other methods, but also represent strong generalizabilities crossing different urban datasets. The chapter 3 is supported by the conference publication at IJCAI-2020 [19].

Chapter 4 proposes we propose two data-driven models for CFD prediction on the Sydney Trains rail network. The first model, called OLS-AO, is based on average optimization and historic guidance and captures the dynamic changes in latent attributes over time. To improve prediction accuracy, we further designed another OLS model, called OLS-MR, to tackle sudden changes in CFDs. Intensive experiments show that our proposed methods outperform several baselines. A dual-track strategy, which combines both OLS-AO and OLS-MR, achieves the best results on weekdays, and OLS-AO is the most outstanding method for weekend predictions. The chapter 4 is supported by the journal and conference publications at TKDE and CIKM-2018, respectively [21, 22].

Chapter 5 targets at the potential passenger flow prediction, which is a novel study that brings benefits to the urban transportation development. To address this spatio-temporal problem, we design a multi-view localized correlation learning model (MLC-PPF) for the PPF prediction. The k-nearest indicator matrix H is constructed by the real geographical neighbors and regional properties. MLC-PPF can learn the correlations between each known area and its k-nearest neighbors with the cross-domain knowledge guidance. We evaluate the performance of our method with a set of well-designed experiments. The chapter 5 is supported by the conference publication at AAAI-2020 [20].

6.2 Future Work

Most recent studies, such as [64, 81, 108], predict the OD matrix based on a deep learning model via graph convolution strategy or recurrent neural network. For example, [81] divides the city area into grids and predicts the passenger demands of given origin grid-areas and destination grid-areas at a given time slot. However, it seems that this method does not consider the *real-time delayed data collection* problem, since it makes all trajectory data observed when testing. At current research stage, deep learning based methods are difficult to solve the real online problem. Therefore, our further effort will be made towards developing a real online model based on deep learning.

BIBLIOGRAPHY

[1] Acar, E., Dunlavy, D. M., Kolda, T. G., and Mørup, M. (2011).
Scalable tensor factorizations for incomplete data.
Chemometrics and Intelligent Laboratory Systems, 106(1):41–56.

[2] Alexander, L., Jiang, S., Murga, M., and González, M. C. (2015).
Origin–destination trips by purpose and time of day inferred from mobile phone data.
Transportation research part c: emerging technologies, 58:240–250.

[3] Ashok, K. and Ben-Akiva, M. E. (2002).
Estimation and prediction of time-dependent origin-destination flows with a stochastic mapping to path flows and link flows.
Transportation science, 36(2):184–198.

[4] Atluri, G., Karpatne, A., and Kumar, V. (2017).
Spatio-temporal data mining: A survey of problems and methods.
arXiv:1711.04710.

[5] Atluri, G., Karpatne, A., and Kumar, V. (2018).
Spatio-temporal data mining: A survey of problems and methods.
ACM Computing Surveys (CSUR), 51(4):83.

[6] Blondel, M., Kubo, Y., and Naonori, U. (2014).
Online passive-aggressive algorithms for non-negative matrix factorization and completion.
In *Artificial Intelligence and Statistics*, pages 96–104.

[7] Candès, E. J. and Recht, B. (2009).
Exact matrix completion via convex optimization.
Foundations of Computational mathematics, 9(6):717.

[8] Cao, B., Shen, D., Sun, J.-T., Wang, X., Yang, Q., and Chen, Z. (2007).

Detect and track latent factors with online nonnegative matrix factorization.
In *IJCAI*, volume 7, pages 2689–2694.

[9] Castro-Neto, M., Jeong, Y.-S., Jeong, M.-K., and Han, L. D. (2009).
Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions.
Expert systems with applications, 36(3):6164–6173.

[10] Ceapa, I., Smith, C., and Capra, L. (2012).
Avoiding the crowds: understanding tube station congestion patterns from trip data.
In *Proceedings of the ACM SIGKDD international workshop on urban computing*, pages 134–141. ACM.

[11] Chen, F.-W. and Liu, C.-W. (2012).
Estimation of the spatial rainfall distribution using inverse distance weighting (idw) in the middle of taiwan.
Paddy and Water Environment, 10(3):209–222.

[12] Chen, M.-C. and Wei, Y. (2011).
Exploring time variants for short-term passenger flow.
Journal of Transport Geography, 19(4):488–498.

[13] Cheng, S. and Lu, F. (2017).
A two-step method for missing spatio-temporal data reconstruction.
ISPRS International Journal of Geo-Information, 6(7):187.

[14] Chua, F. C. T., Oentaryo, R. J., and Lim, E.-P. (2013).
Modeling temporal adoptions using dynamic matrix factorization.
In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 91–100. IEEE.

[15] Deng, D., Shahabi, C., Demiryurek, U., Zhu, L., Yu, R., and Liu, Y. (2016a).
Latent space model for road networks to predict time-varying traffic.
In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1525–1534. ACM.

[16] Deng, D., Shahabi, C., Demiryurek, U., Zhu, L., Yu, R., and Liu, Y. (2016b).
Latent space model for road networks to predict time-varying traffic.
In *Proceedings of the 22nd ACM SIGKDD*, pages 1525–1534.

[17] Elkahky, A. M., Song, Y., and He, X. (2015).
A multi-view deep learning approach for cross domain user modeling in recommendation systems.
In *Proceedings of the 24th International Conference on World Wide Web*, pages 278–288. International World Wide Web Conferences Steering Committee.

[18] Gao, S., Luo, H., Chen, D., Li, S., Gallinari, P., and Guo, J. (2013).
Cross-domain recommendation via cluster-level latent factor model.
In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 161–176. Springer.

[19] Gong, Y., Li, Z., Zhang, J., Liu, W., Chen, B., and Dong, X. (2020a).
A spatial missing value imputation method for multi-view urban statistical data.
In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 1310–1316.

[20] Gong, Y., Li, Z., Zhang, J., Liu, W., and Yi, J. (2020b).
Potential passenger flow prediction: A novel study for urban transportation development.
In *Proceedings of 34th Conference on AAAI*, pages 4020–4027.

[21] Gong, Y., Li, Z., Zhang, J., Liu, W., and Zheng, Y. (2020c).
Online spatio-temporal crowd flow distribution prediction for complex metro system.
IEEE Transactions on Knowledge and Data Engineering.

[22] Gong, Y., Li, Z., Zhang, J., Liu, W., Zheng, Y., and Kirsch, C. (2018).
Network-wide crowd flow prediction of sydney trains via customized online non-negative matrix factorization.
In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1243–1252.

[23] Guo, S., Lin, Y., Feng, N., Song, C., and Wan, H. (2019).
Attention based spatial-temporal graph convolutional networks for traffic flow forecasting.
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 922–929.

[24] Gupta, M. D. and Xiao, J. (2011).

Non-negative matrix factorization as a feature selection tool for maximum margin classifiers.
In *CVPR 2011*, pages 2841–2848. IEEE.

[25] Hazelton, M. L. (2003).
Some comments on origin–destination matrix estimation.
Transportation Research Part A: Policy and Practice, 37(10):811–822.

[26] He, X. and Niyogi, P. (2004).
Locality preserving projections.
In *Advances in neural information processing systems*, pages 153–160.

[27] Hofleitner, A., Herring, R., and Bayen, A. (2012).
Probability distributions of travel times on arterial networks: a traffic flow and horizontal queuing theory approach.
In *91st Transportation Research Board Annual Meeting*, number 12-0798.

[28] Hsieh, H.-P., Lin, S.-D., and Zheng, Y. (2015).
Inferring air quality for station location recommendation based on urban big data.
In *the 21th ACM SIGKDD International Conference*.

[29] Hu, L., Cao, J., Xu, G., Cao, L., Gu, Z., and Zhu, C. (2013).
Personalized recommendation via cross-domain triadic factorization.
In *Proceedings of the 22nd international conference on World Wide Web*, pages 595–606. ACM.

[30] Hu, L., Cao, L., Cao, J., Gu, Z., Xu, G., and Yang, D. (2016).
Learning informative priors from heterogeneous domains to improve recommendation in cold-start user domains.
ACM Transactions on Information Systems (TOIS), 35(2):13.

[31] Jing, P., Su, Y., Jin, X., and Zhang, C. (2018).
High-order temporal correlation model learning for time-series prediction.
IEEE transactions on cybernetics, (99):1–13.

[32] Kolda, T. G. and Bader, B. W. (2009).
Tensor decompositions and applications.
SIAM review, 51(3):455–500.

[33] Lam, X. N., Vu, T., Le, T. D., and Duong, A. D. (2008).
Addressing cold-start problem in recommendation systems.
In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pages 208–211. ACM.

[34] Lambiotte, R., Delvenne, J.-C., and Barahona, M. (2008).
Laplacian dynamics and multiscale modular structure in networks.
arXiv preprint arXiv:0812.1770.

[35] Lee, D. D. and Seung, H. S. (2001).
Algorithms for non-negative matrix factorization.
In *Advances in neural information processing systems*, pages 556–562.

[36] Leng, B., Zeng, J., Xiong, Z., Lv, W., and Wan, Y. (2013).
Probability tree based passenger flow prediction and its application to the beijing subway system.
Frontiers of Computer Science, 7(2):195–203.

[37] Li, L., Su, X., Wang, Y., Lin, Y., Li, Z., and Li, Y. (2015).
Robust causal dependence mining in big data network and its application to traffic flow predictions.
Transportation Research Part C: Emerging Technologies, 58:292–307.

[38] Li, X., Cui, G., and Dong, Y. (2016).
Graph regularized non-negative low-rank matrix factorization for image clustering.
IEEE transactions on cybernetics, 47(11):3840–3853.

[39] Li, Y., Wang, X., Sun, S., Ma, X., and Lu, G. (2017).
Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks.
Transportation Research Part C: Emerging Technologies, 77:306–328.

[40] Li, Y., Yang, M., and Zhang, Z. M. (2018).
A survey of multi-view representation learning.
IEEE Transactions on Knowledge and Data Engineering.

[41] Lin, L., Li, J., Chen, F., Ye, J., and Huai, J. (2017).
Road traffic speed prediction: a probabilistic model fusing multi-source data.
IEEE Transactions on Knowledge and Data Engineering, 30(7):1310–1323.

[42] Liu, L. and Chen, R.-C. (2017).
A novel passenger flow prediction model using deep learning methods.
Transportation Research Part C: Emerging Technologies, 84:74–91.

[43] 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.

[44] Liu, L., Zhang, R., Peng, J., Li, G., Du, B., and Lin, L. (2018a).
Attentive crowd flow machines.
arXiv preprint arXiv:1809.00101.

[45] Liu, R., Li, S., Yang, L., and Yin, J. (2018b).
Energy-efficient subway train scheduling design with time-dependent demand based on an approximate dynamic programming approach.
IEEE Transactions on Systems, Man, and Cybernetics: Systems, (99):1–16.

[46] Liu, X., Li, M., Wang, L., Dou, Y., Yin, J., and Zhu, E. (2017).
Multiple kernel k-means with incomplete kernels.
In *Proceedings of 27th Conference on Association for the Advancement of Artificial Intelligence*, pages 2259–2265.

[47] Luo, X., Wu, H., Yuan, H., and Zhou, M. (2019).
Temporal pattern-aware qos prediction via biased non-negative latent factorization of tensors.
IEEE transactions on cybernetics.

[48] Ma, W. and Qian, Z. S. (2018).
Statistical inference of probabilistic origin-destination demand using day-to-day traffic data.
Transportation Research Part C: Emerging Technologies, 88:227–256.

[49] Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., and Wang, Y. (2017).
Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction.
Sensors, 17(4):818.

[50] Ma, X., Zhang, J., Du, B., Ding, C., and Sun, L. (2018).

Parallel architecture of convolutional bi-directional lstm neural networks for network-wide metro ridership prediction.
IEEE Transactions on Intelligent Transportation Systems, 20(6):2278–2288.

[51] Ma, Y., Lin, T., Cao, Z., Li, C., Wang, F., and Chen, W. (2016).
Mobility viewer: An eulerian approach for studying urban crowd flow.
IEEE Transactions on Intelligent Transportation Systems, 17(9):2627–2636.

[52] Milenkovic, M., Bojovic, N., Macura, D., and Nuhodzic, R. (2013).
Kalman filtering applied to forecasting the demand for railway passenger services.

[53] Mohamed, K., Côme, E., Oukhellou, L., and Verleysen, M. (2017).
Clustering smart card data for urban mobility analysis.
IEEE Transactions on Intelligent Transportation Systems, 18(3):712–728.

[54] Murgante, B. and Danese, M. (2011).
Urban versus rural: the decrease of agricultural areas and the development of urban zones analyzed with spatial statistics.
International Journal of Agricultural and Environmental Information Systems, 2(2):16–28.

[55] Ni, M., He, Q., and Gao, J. (2017).
Forecasting the subway passenger flow under event occurrences with social media.
IEEE Transactions on Intelligent Transportation Systems, 18(6):1623–1632.

[56] Oneto, L., Fumeo, E., Clerico, G., Canepa, R., Papa, F., Dambra, C., Mazzino, N., and Anguita, D. (2017).
Dynamic delay predictions for large-scale railway networks: Deep and shallow extreme learning machines tuned via thresholdout.
IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(10):2754–2767.

[57] Pan, L. and Li, J. (2010).
K-nearest neighbor based missing data estimation algorithm in wireless sensor networks.
Wireless Sensor Network, 2(2):115.

[58] Pan, Z., Liang, Y., Wang, W., Yu, Y., Zheng, Y., and Zhang, J. (2019).
Urban traffic prediction from spatio-temporal data using deep meta learning.
In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1720–1730. ACM.

[59] Pelletier, M.-P., Trépanier, M., and Morency, C. (2011).
Smart card data use in public transit: A literature review.
Transportation Research Part C: Emerging Technologies, 19(4):557–568.

[60] Petersen, K. B., Pedersen, M. S., et al. (2008).
The matrix cookbook.
Technical University of Denmark, 7(15):510.

[61] Ran, B., Song, L., Zhang, J., Cheng, Y., and Tan, H. (2016).
Using tensor completion method to achieving better coverage of traffic state estimation from sparse floating car data.
PloS one, 11(7):e0157420.

[62] Ranjbar, M., Moradi, P., Azami, M., and Jalili, M. (2015).
An imputation-based matrix factorization method for improving accuracy of collaborative filtering systems.
Engineering Applications of Artificial Intelligence, 46:58–66.

[63] Rasmussen, C. E. and Williams, C. K. (2006).
Gaussian processes for machine learning, volume 1.
MIT press Cambridge.

[64] Ren, J. and Xie, Q. (2017).
Efficient od trip matrix prediction based on tensor decomposition.
In *2017 18th IEEE International Conference on Mobile Data Management (MDM)*, pages 180–185. IEEE.

[65] Rendle, S., Balby Marinho, L., Nanopoulos, A., and Schmidt-Thieme, L. (2009).
Learning optimal ranking with tensor factorization for tag recommendation.
In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 727–736. ACM.

[66] Singh, A. P. and Gordon, G. J. (2008).
Relational learning via collective matrix factorization.
In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 650–658. ACM.

[67] Su, X. and Khoshgoftaar, T. M. (2009).
A survey of collaborative filtering techniques.
Advances in artificial intelligence, 2009.

[68] Sun, L., Lu, Y., Jin, J. G., Lee, D.-H., and Axhausen, K. W. (2015a).
An integrated bayesian approach for passenger flow assignment in metro networks.
Transportation Research Part C: Emerging Technologies, 52:116–131.

[69] 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.

[70] Sun, Y., Leng, B., and Guan, W. (2015b).
A novel wavelet-svm short-time passenger flow prediction in beijing subway system.
Neurocomputing, 166:109–121.

[71] Tan, H., Feng, G., Feng, J., Wang, W., Zhang, Y.-J., and Li, F. (2013a).
A tensor-based method for missing traffic data completion.
Transportation Research Part C: Emerging Technologies, 28:15–27.

[72] Tan, H., Wu, Y., Feng, G., Wang, W., and Ran, B. (2013b).
A new traffic prediction method based on dynamic tensor completion.
Procedia-Social and Behavioral Sciences, 96:2431–2442.

[73] Taneja, A. and Arora, A. (2018).
Cross domain recommendation using multidimensional tensor factorization.
Expert Systems with Applications, 92:304–316.

[74] Thida, M., Eng, H.-L., and Remagnino, P. (2013).
Laplacian eigenmap with temporal constraints for local abnormality detection in crowded scenes.
IEEE Transactions on Cybernetics, 43(6):2147–2156.

[75] Tobler, W. R. (1970).
A computer movie simulating urban growth in the detroit region.
Economic geography, 46(sup1):234–240.

[76] Trivedi, A., Rai, P., Daumé III, H., and DuVall, S. L. (2010).
Multiview clustering with incomplete views.
In *The workshop pf 24th Conference on Neural Information Processing Systems*, pages 1–7.

[77] Van Buuren, S. (2018).
Flexible imputation of missing data.

CRC press.

[78] Van Der Hurk, E., Kroon, L., Maróti, G., and Vervest, P. (2015).
Deduction of passengers' route choices from smart card data.
IEEE Transactions on Intelligent Transportation Systems, 16(1):430–440.

[79] Van Der Voort, M., Dougherty, M., and Watson, S. (1996).
Combining kohonen maps with arima time series models to forecast traffic flow.
Transportation Research Part C: Emerging Technologies, 4(5):307–318.

[80] Wang, F., Tan, C., Li, P., and König, A. C. (2011).
Efficient document clustering via online nonnegative matrix factorizations.
In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 908–919. SIAM.

[81] Wang, Y., Yin, H., Chen, H., Wo, T., Xu, J., and Zheng, K. (2019).
Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling.
In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1227–1235. ACM.

[82] Wang, Y., Zheng, Y., and Xue, Y. (2014).
Travel time estimation of a path using sparse trajectories.
In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 25–34. ACM.

[83] Wei, Y. and Chen, M.-C. (2012).
Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks.
Transportation Research Part C: Emerging Technologies, 21(1):148–162.

[84] Williams, B. M. and Hoel, L. A. (2003).
Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results.
Journal of transportation engineering, 129(6):664–672.

[85] Wu, T. and Li, Y. (2013).
Spatial interpolation of temperature in the united states using residual kriging.
Applied Geography, 44:112–120.

[86] Xiong, L., Chen, X., Huang, T.-K., Schneider, J., and Carbonell, J. G. (2010). Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 211–222. SIAM.

[87] Xu, C., Tao, D., and Xu, C. (2013). A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*.

[88] Xu, C., Tao, D., and Xu, C. (2015). Multi-view learning with incomplete views. *IEEE Transactions on Image Processing*, 24(12):5812–5825.

[89] Xu, D.-w., Wang, Y.-d., Jia, L.-m., Qin, Y., and Dong, H.-h. (2017). Real-time road traffic state prediction based on arima and kalman filter. *Frontiers of Information Technology & Electronic Engineering*, 18(2):287–302.

[90] Xu, M., Xie, X., Lv, P., Niu, J., Wang, H., Li, C., Zhu, R., Deng, Z., and Zhou, B. (2019). Crowd behavior simulation with emotional contagion in unexpected multihazard situations. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

[91] Ye, Q., Szeto, W. Y., and Wong, S. C. (2012). Short-term traffic speed forecasting based on data recorded at irregular intervals. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1727–1737.

[92] Yi, X., Zheng, Y., Zhang, J., and Li, T. (2016a). St-mvl: filling missing values in geo-sensory time series data. In *Proceedings of 25th International Joint Conference on Artificial Intelligence*, pages 2704–2710.

[93] Yi, X., Zheng, Y., Zhang, J., and Li, T. (2016b). St-mvl: filling missing values in geo-sensory time series data.

[94] Yu, B., Liu, Y., and Sun, Q. (2016). A content-adaptively sparse reconstruction method for abnormal events detection with low-rank property. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(4):704–716.

[95] Zhan, X., Zheng, Y., Yi, X., and Ukkusuri, S. V. (2016).
Citywide traffic volume estimation using trajectory data.
IEEE Transactions on Knowledge and Data Engineering, 29(2):272–285.

[96] Zhang, J., Zheng, Y., and Qi, D. (2017).
Deep spatio-temporal residual networks for citywide crowd flows prediction.
In *AAAI*, pages 1655–1661.

[97] 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.

[98] Zhang, J., Zheng, Y., Sun, J., and Qi, D. (2019).
Flow prediction in spatio-temporal networks based on multitask deep learning.
IEEE Transactions on Knowledge and Data Engineering.

[99] Zhang, Y. and Yeung, D.-Y. (2012).
Overlapping community detection via bounded nonnegative matrix tri-factorization.
In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 606–614. ACM.

[100] Zhao, J., Qu, Q., Zhang, F., Xu, C., and Liu, S. (2017).
Spatio-temporal analysis of passenger travel patterns in massive smart card data.
IEEE Transactions on Intelligent Transportation Systems.

[101] Zhao, J., Zhang, F., Tu, L., Xu, C., Shen, D., Tian, C., Li, X.-Y., and Li, Z. (2016).
Estimation of passenger route choice pattern using smart card data for complex metro systems.
IEEE Transactions on Intelligent Transportation Systems, 18(4):790–801.

[102] Zheng, Y. (2015).
Methodologies for cross-domain data fusion: An overview.
IEEE transactions on big data, 1(1):16–34.

[103] Zheng, Y., Capra, L., Wolfson, O., and Yang, H. (2014).
Urban computing: concepts, methodologies, and applications.
ACM Transactions on Intelligent Systems and Technology (TIST), 5(3):38.

[104] Zheng, Y., Liu, F., and Hsieh, H.-P. (2013).
U-air: When urban air quality inference meets big data.

In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1436–1444. ACM.

[105] Zheng, Y., Liu, Y., Yuan, J., and Xie, X. (2011).
Urban computing with taxicabs.
In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 89–98. ACM.

[106] Zhou, J. and Huang, Z. (2018).
Recover missing sensor data with iterative imputing network.
In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*.

[107] Zhou, J. and Tung, A. K. (2015).
Smiler: A semi-lazy time series prediction system for sensors.
In *2015 ACM SIGMOD*, pages 1871–1886.

[108] Zhou, X. and Mahmassani, H. S. (2006).
Dynamic origin-destination demand estimation using automatic vehicle identification data.
IEEE Transactions on intelligent transportation systems, 7(1):105–114.