**Title**

Statistical Learning for High-Dimensional Networked Data in Transportation Systems



**Author**

Sun, Ran

**Publication Date**

2023

Peer reviewed|Thesis/dissertation

Statistical Learning for High-Dimensional Networked Data in Transportation Systems

By

RAN SUN

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Civil and Environmental Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

_____

Yueyue Fan, Co-Chair

_____

James Sharpnack, Co-Chair

_____

H. Michael Zhang

_____

Giovanni Circella

Committee in Charge

2023

*To the sunshine, for the warmth and golden rays,*

*To the moonlight, for the peace and silver gaze.*

# Contents

# Abstract

Understanding the mobility pattern and network flow provides fundamental knowledge for decision making in transportation planning and operations. These insights play a critical role in travel demand modeling, traffic management and control, and processes in the development of robust and sustainable transportation systems. There has been a long history of developing estimation methods to better understand networked data. However, direct observations often suffer from several challenges: high-dimensionality, limited coverage, and low fidelity. Furthermore, existing literature tends to separate data with data-driven methods and domain knowledge with behavioral-based and physics-informed models.

This dissertation focuses on statistical learning on high-dimensional networked data in transportation. We consider signal from direct observations and the interdependence among various components in the network, under the regularization of the domain knowledge and structural information. The goal of the research is to develop a methodology to improve the quality, efficiency, and robustness of the estimation of high-dimensional networked data in transportation systems. Based on the interdependency of data from various network levels, the development of the statistical learning frameworks targets the following specific key objectives : 1) estimating unknown OD demand from observable link flow in the networks; 2) learning meaningful data representations in networks for critical information extraction and anomaly detection; and 3) statistical filtering for data on directed graphs.

In OD estimation, we present a modeling framework for OD demand estimation based on observed traffic flow data in a transportation network, from a fresh angle of stochastic programming. The proposed two-stage stochastic programming method is flexible for incorporating various design principles and risk preferences into domain knowledge regarding travel behavioral and physical rules. Besides, a benefit comes from the scenario representation, where

the point estimate can be combined with estimation of the discrete approximation to the demand distribution. We demonstrate that under the proposed framework, well-established theories and methods for stochastic programming, including epi-convergence and scenario-decomposition, can be exploited to advance the analytical and computational capability of the estimation model. The applicability and efficiency of the proposed method are illustrated via numerical examples based on highway and transit networks of various sizes.

In representation learning, we introduce a new perspective that the critical information of the data should reflect how the data is used in downstream applications, which carries a different research design philosophy adopted in most existing data representation learning methods. We propose an application-driven representation learning framework by incorporating information loss for the downstream application into the data encoding-decoding process. The proposed approach is formulated as a Stiefel manifold optimization problem. The effectiveness of the proposed framework is demonstrated through three case studies: transportation network performance assessment, vehicular emission estimation, and anomaly detection of travel demand. Experiments show that our proposed approach performs better than classical representation learning approaches, especially in applications involving complex network interdependence.

In graph signal filtering, we developed a Continuous-time Markov chain based filtering methods on directed graphs using nonparametric regression technique. Through bridging the stochastic process and algebraic graph theory, we utilize the transition matrix basis that is dependent on the connectivity of points across varied density regions. Compared with methods dedicated to undirected graph and spatial filtering methods, our approach is capable of capturing directional information flows and local asymmetric structure in data observations. The performance of our approach is evidenced through a series of synthetic and real-world case studies on network traffic flow. This work demonstrates the potential for incorporating heterogeneous structures for data defined on irregular domains.

# Acknowledgments

My time in Davis has been an amazing ride. First and foremost, I would like to express my deepest gratitude to my advisor, Professor Yueyue Fan, for her continuous help, guidance, and encouragement to explore the boundaries of my research field. Her broad understanding of the field has paved the way for the completion of my dissertation. Beyond knowledge, her wisdom will continue to inspire me in my future endeavors.

I would like to thank my co-advisor, Professor James Sharpnack. His exceptional insights on a very diverse range of topics in statistics and machine learning have encouraged me to learn and explore new directions and opportunities. His rigorous and diligent working attitude served as a foundational model for my own research approach.

My sincere thanks also go out to Professor Giovanni Circella for supporting many years of my Ph.D. study. He sets the perfect example for a passionate thinker, an insightful mentor, and a caring friend. While working with him, I am fortunate to have been exposed to a wide range of topics across the transportation field, which improved my understanding on demand and behavioral modeling research and practices in many other fields. I want to thank Professor Michael Zhang for his education and guidance, especially regarding traffic flow theory and network modeling, which have been some of the best memories and lessons that I will never forget.

I also want to extend my gratitude to other mentors and collaborators, including Professor Miguel Jaller for the help in qualifying process and education in travel demand, supply chain and logistics, Professor Susan Handy on travel behavior and planning.

I am also grateful for all my friends, lab mates and colleagues at UC Davis, Han Yang, Xiaodong Qian, Yudi Yang, Xinyue Hu, Sarah Grajdura, Junia Compostella, Keita Makino, Guozhen Li, Jinpeng Gao, for providing valuable suggestions, guidance, and a stimulating and fun environment.

I am extremely fortunate to be accompanied by my wife, Tianjing Zhao. She has been our CEO (Chief Encouragement Officer), as nothing could beat her uplifting attitudes. She

has been an undisputed 'Captain of the Stress Diffuser'. If patience had a face, it would undoubtedly resemble hers. And most importantly, she has been my unrivaled chef, for chiffon cake and pan-seared pot stickers.

Finally, I want to express the most sincere appreciation to my parents for their unwavering support for my studies and career pursuit. I would never go this far without their unconditional love and commitment.

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation and Background

Mobility, travel demand, and traffic flow have been some of the major areas of focus in transportation science for many years. Travel demand information serves as a critical input in almost all levels of mobility planning, design, and control applications. The increasing prevalence of ride-sharing and car-sharing services further necessitates the development of statistical tools for understanding demand and network flow. Ideally, travel demand should tell individuals' travel needs from an origin to a destination during a certain time period. However, due to financial and mobility constraints imposed on travelers, not all potential travel can be realized. Unlike travel demand information, which is not usually vastly available to planners and decision makers, traffic flow information is more accessible thanks to the advancement in sensing technologies. As the realization of the interactions of travel demand and infrastructure supply, traffic flow directly reveals the state of transportation systems, enabling various real-time traffic management and control strategies.

Opportunities come with significant challenges when it comes to the management, processing, storage, and utilization of large-scale high-dimensional transportation data (Vlahogianni, 2015; Cuzzocrea, 2019). Some of the major challenges lie in the aspects on how

to mitigate the uncertainties within these raw data and effectively utilize them to extract useful information that can be carried further to down stream applications for planning and operational purposes. Here we summarize three main challenges:

- **High dimensionality:** Understanding complex spatio-temporal pattern of transportation data is challenging as transportation systems are often highly stochastic and transportation related data usually live in high dimensional space. In addition, measurements from different views, such as link flow, demand, vehicle ID tracking, path level trajectory, pose uncertainties in data fusion and inference from indirect measurements. Conventional approaches may not be directly applicable to high-dimensional transportation systems.
- **Limited coverage and partial view:** Fixed location sensors, despite having high temporal resolution, can only capture a partial picture of the transportation system spatially. They are limited in their ability to provide comprehensive data across different areas. On the other hand, mobile sensors have the potential to measure both the spatial and temporal dimensions of the system more effectively. However, they are more expensive to deploy widely and are prone to noise and fluctuations.
- **Noise and low fidelity:** Observations in transportation data are rarely free of noise. The presence of noise and fluctuations from various sources poses challenges in building models that can generalize well to real-world scenarios. Furthermore, the lack of regular maintenance of sensors and infrastructure can result in missing data, further complicating the analysis and interpretation of transportation data. Addressing the noise and low-fidelity issues requires more robust methods.

In transportation science community, there are mainly two categories of philosophy in modeling the transportation systems: model-based approaches and data-driven approaches. Tremendous effort has been dedicated to developing realistic physical-based or behavior-based models that can capture the mechanism of transportation systems. For example,

the widely used BRP volume-delay function (Bureau of Public Roads, 1964) measures the relationship between travel flow and link travel time. Car following models can be used to capture the microscopic flow dynamics (Newell, 1961; Gipps, 1981). Traffic assignment models translate the travel demand to traffic flow, given the cost function and network supply (Daganzo and Sheffi, 1977; Shao et al., 2006). Nested logit models capture the behavioral decisions on travel mode choice (Wen and Koppelman, 2001).

In another school of thoughts, data-driven approaches has gained increasing attention as information technologies advance recently. For example, wavelet transform is used to detect the characteristics of traffic state and bottlenecks (Zheng et al., 2011). For data measured in transportation networks, several studies use spatio-temporal techniques to investigate different aspects of link-based traffic flow, including volume (Tan et al., 2016; Ran et al., 2016; Tan et al., 2013), speed (Goulart et al., 2017; Asif et al., 2016) and accidents (Ke et al., 2019). For spatio-temporal mobility pattern mining,Sun and Axhausen (2016) and Wang et al. (2021) developed decomposition methods that can discover latent variables dominating the interactions of demand and supply.

In some cases, the model assumptions are not realistic given the high uncertainty and stochasticity of the systems. On the other hand, data-driven methods may suffer from overfitting to noise and lack of interpretability and generalizability. In this dissertation, we incorporate model-based domain knowledge into estimation tasks, leveraging the underlying physics and variations caused by spatial and temporal uncertainties. We investigate some fundamental challenges in noisy high-dimensional networked data in transportation systems using optimization and statistical learning tools. We hope to bridge several gaps in discovering the underlying phenomenon of high-dimensional networked traffic flow. Specifically, based on the interdependency of data on various levels of network components, we try to answer three main questions for high-dimensional networked data:

1. Can we benefit from multiple sources of data in high-dimensional networked flow estimation?

2. Can we extract representative critical information from high-dimensional data in transportation networked systems?

3. Can we leverage structural information to inform high-dimensional estimations?

## 1.2 Research Objectives and Significance

Although a fair amount of research has been dedicated to understanding networked data using both model-based and data-driven approaches, the integration of domain-specific knowledge, structural information, and latent data patterns has not received enough attention. Domain-specific knowledge, for instance, can provide a valuable contextual understanding of the underlying phenomenon of the observations. It often contributes model-based insights to the conceptualization of the estimation steps. Similarly, structural information provides crucial regularization or constraints that help restrict the feasibility region of the potential solution. Latent patterns in the data can uncover hidden relationships and complex interactions that may not be immediately evident from the raw observations. These patterns can often offer invaluable information on the generative mechanism of networked data, leading to more fundamental insights.

In this dissertation, we classify the various components of the transportation system by their generative mechanism in the networks, as link-based data, node-based data, path-based data and node-to-node-based data. Node-based and link-based data are among the most widely available types. These include microscopic traffic flow density, flow rate, and speed. Path-based data are becoming more prevalent due to advances in sensing technology, including GPS traces for travelers and high-resolution lane-level vehicle trajectory. Node-to-node data, including OD demand, often require special treatment because of the highly correlated underlying decision-making processes. In Figure 1.1, we present an illustration of various types of networked data and their relationships. Travel Demand and infrastructure supply communicate through the underlying directed network. The observations of various levels in the network are consequences of the steady and transition states of the demand and

supply. By considering the data generation mechanism and the network operation knowledge, we can provide better understandings of the network systems. Based on the connection of the information with the networks, we identify three main research tasks, which we now explain here. This dissertation aims to address three research gaps through the pursuit of various

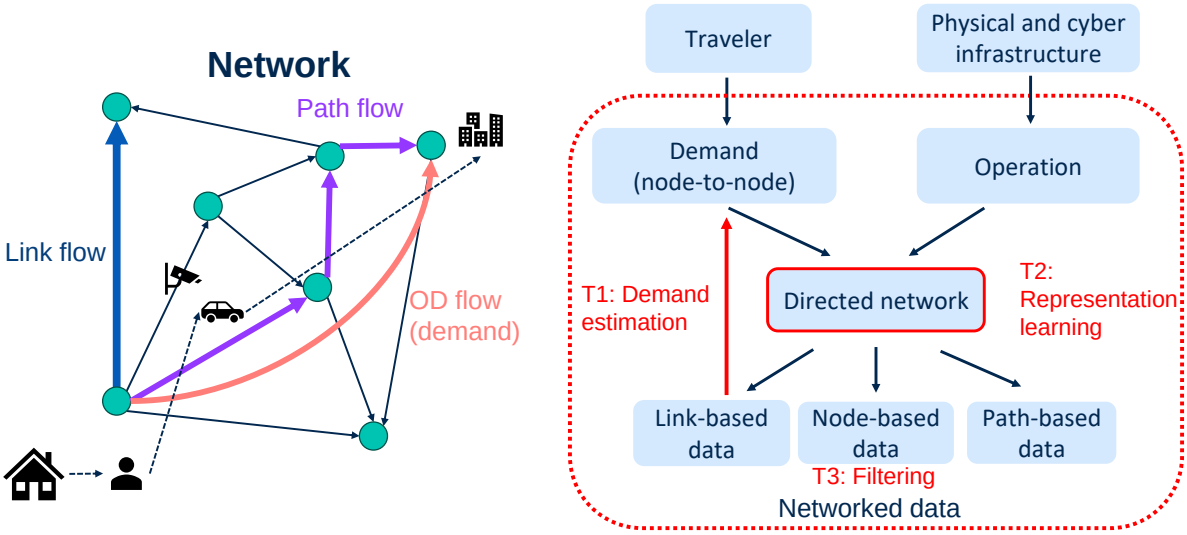



Figure 1.1: Research framework

optimization and statistical machine learning methodologies, with a focus on estimation tasks in transportation networks. Among the broad application opportunities, we demonstrate the capability of purposed methods through understanding the characteristics of networked traffic flow, from Origin-Destination (OD) demand to link/path flow, vehicle trajectories, and others. We focus on different types of observable networked data for different tasks, and they are connected through the underlying directed network structure and network operation domain knowledge. Specifically, we are interested in these tasks:

1. Estimating OD demand from link flow observations (node-based to link-based information),

2. Discover latent patterns and critical information in networked data (connection between information at various levels),

3. Noise filtering for flow on directed graphs (node-based information).

Although these tasks are directly aimed at different levels of information, the network structure and domain knowledge remain the underlying theme that connects various objectives.

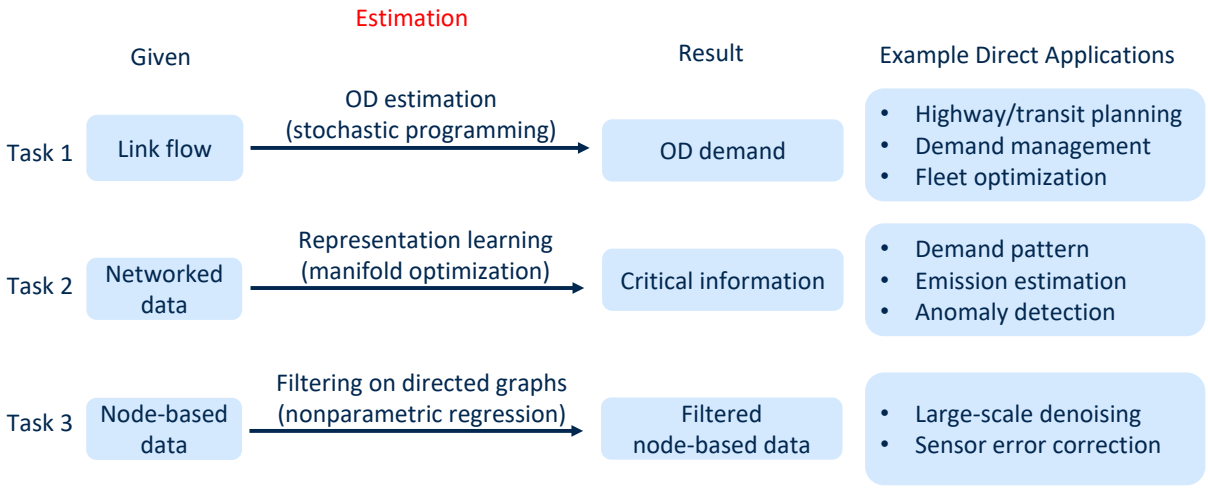



Figure 1.2: Research Tasks

These tasks are shown in Figure 1.2. Specifically, we start by developing a novel stochastic programming OD estimation method which bridges the philosophy of statistical inference and stochastic programming. We also develop an application-driven representation learning combining the intrinsic structure of the data and the complex relationship that dominates the data generation mechanism. Furthermore, we develop a statistical filtering method on directed graphs that can take advantage of the structural information and the varied levels of local correlation in the observations.

These objectives are centered on the decomposition principle, where large-scale high-dimensional data can be decomposed into more meaningful or more manageable pieces to

draw conclusions based on. They would help uncover the underlying pattern of daily traffic evolution in the network setting. The learned critical signals and latent patterns could further benefit much broader operation and planning tasks.

## 1.3 Dissertation Organization

The structure of this dissertation is as follows.

**Chapter 2: Stochastic OD Demand Estimation Using Stochastic Programming** This chapter aims to address the OD estimation problem based on link flow counts. We approach the problem from a fresh angle based on two-stage stochastic programming (SP) framework, where demand parameter estimation is treated as the first stage decision and demand reconstruction as the scenario-dependent recourse decision. The performance of the proposed framework is demonstrated for both highway and transit networks.

**Chapter 3: Application-Driven Representation Learning for Feature Extraction and Anomaly Detection in Transportation Systems** This chapter deals with the problem of finding meaningful representation of high-dimensional transportation flow data. A new method of finding dominant features is established through incorporating both information in data and structure in downstream applications. We show that the application metric can help with revealing hidden structure in high-dimensional raw observations, through a series of tasks, including feature extraction of mobility data, emission estimation based vehicle trajectory and anomaly detection on travel demand.

**Chapter 4: Continuous-time Markov Chain based Filtering on Directed Graphs** We propose Continuous-time Markov Chain (CTMC) based filtering as a novel approach for high-dimensional estimation on directed graphs from noisy observations. We approach the graph signal filtering problem under the synthesis framework by leveraging a continuous-time transition basis. The method can be easily applied to the cases where the graph structures are given as prior or to the more general applications by leveraging the KNN or proximity graphs. Compared with standard methods dedicated to undirected graphs and spatial infor-

mation, we show that the CTMC filtering developed for directed graph can reveal asymmetric structure and can generate varied levels of adaptivity of data observations.

## 1.4 Summary of Contribution

This dissertation is primarily dedicated to the methodological and theoretical improvement of high-dimensional estimation methods and is readily generalizable to various real-world applications. Although the context and case studies of this dissertation are networked traffic flow, its benefits go far beyond what is experimented here. Our proposed methods are capable of capturing complex interactions among high-dimensional observed data and revealing critical information through hidden structures discovered. Here we list boarder aspects of methodological and practical contributions

- A multistage estimation framework leveraging multiple observation sets that exploits structural information in data,
- A scalable decomposition solution for large-scale high-dimensional estimation task,
- A representation learning and feature extraction method that incorporate information in both data and downstream applications,
- A dimensionality reduction based trajectory level emission estimation method,
- An anomaly detection method based on latent structure within the data observations,
- A nonparametric regression based high-dimensional filtering on directed graphs.

# Chapter 2

# Stochastic OD Demand Estimation Using Stochastic Programming

## 2.1 Introduction

Travel demand plays a critical role in almost all transportation planning and operations decision processes. In the transportation network literature, estimating the network-level Origin Destination (OD) travel demand based on directly measurable local traffic states, such as aggregated link traffic flows, has attracted much attention in recent years. From a system identification perspective, the main challenges involved in those OD estimation problems are 1) translating link-level flow information to the network-level OD demand is not simple, rather it builds on complicated travel behaviors and network physics relations; and 2) link-level flow information alone may not lead to a unique estimation of the OD demand. Consequently, standard statistical techniques for estimation and imputation with limited and indirect data cannot be directly applied; there is a need to explore tailored estimation approaches for the specific context of transportation network level OD demand problem.

Urban transportation system data typically come with high variability across spatial and

temporal dimensions. Considering the uncertain nature of the estimation, we will not discuss previous studies that approached the problem from a deterministic manner and only provided point estimates. Among the many studies on OD estimation that stemmed from statistical estimation approaches, an early paper by Lo et al. (1996) considered using the full likelihood function from link flows based on an independent Poisson distribution assumption. Vardi (1996) further took inter-link dependency into account, while the fixed routing and Markovian routing assumptions are not exactly comparable in a transportation network setting. Hazelton (2000) derived a full likelihood function and applied multivariate normal approximation to simultaneously estimate OD matrix as well as route choice probabilities. Hazelton (2008) took a Bayesian approach with multivariate normal distributed travel demand and provided Bayesian hypothesis testings to measure the precision of estimation. Shao et al. (2015) proposed a weighted least squares approach using first and second order properties of flow observation for multi-class OD demands. Ma and Qian (2018) took an iterative generalized least squares (IGLS) approach to estimate mean and covariance of a probabilistic demand directly from multiple observation sets. Yang et al. (2019) proposed a Generalized Moment Matching (GMM) based estimation framework to infer the probability density function of OD demand using traffic counts in a network. These studies shed light on choices of a conceptual stochastic estimation framework for the demand estimation problem discussed in this study. Within the statistical estimation framework, additional information could be exploited. In the context of network flow estimation, besides the hard data directly collected as filed observations, soft information representing domain knowledge, such as network flow relation and behavioral models, can provide an extra layer of information. Utilization of domain knowledge for transportation network estimation has been reported in the literature. For example, Wang et al. (2016a) investigated network resilience by identifying critical vulnerable links. Liu et al. (2019) tried to understand key nodes in the network using the knowledge from network topology. Wang et al. (2016b) incorporated Stochastic User Equilibrium (SUE) into the estimation for origin-destination (OD) matrix, link choice

proportion, and dispersion parameter. Yang et al. (2018) presented an estimation method that incorporates both transportation operation, route choice and network structure. These studies demonstrated the benefits of using domain knowledge to improve estimation quality for network flows. How to systematically and effectively incorporate various knowledge and information pieces in a statistically meaningful manner remains an active research area.

In a seemingly separate stream of scientific pursuit, we start to see converged thoughts from perspectives of operations research and statistics, mainly stochastic programming and statistical inference in this context. Stochastic programming was initially created as a modeling framework for decision making under uncertainty (Birge and Louveaux, 2011), now with a very broad spectrum of application areas reported in the literature including civil infrastructure (Huang and Loucks, 2000; Liu et al., 2009), supply chain design and management (Santoso et al., 2005; Azaron et al., 2008), process systems (Li and Grossmann, 2021), just to name a few. Beyond the central attention on modeling and computational challenges representing the main-stream OR perspective, several studies paid attention to the statistical properties of the solution to stochastic programs. Among the pioneers, Shapiro (1989) examined the asymptotic of the optimal solution to the stochastic programs and linked it with maximum likelihood estimators. King and Wets (1991) established the epi-consistence for stochastic program with recourse. King and Rockafellar (1993) approached the asymptotic problem using local sensitivity analysis of the generalized equations. Pflug (2003) summarized the consistency, convergence rate, asymptotic distribution and universal bounds for stochastic programs in general. Xu (2010) extended the analysis of the convergence of statistical estimators to the stochastic Nash equilibrium problem and the stochastic generalized equation problem. Later on in the textbook by Shapiro et al. (2021), the statistical properties of sample average approximation estimators and various sampling methods were discussed in great details. These studies opened a door for us to conceptualize and analyze the transportation network estimation problem in the stochastic programming framework.

From a fresh angle using stochastic optimization, we will present an OD estimation ap-

proach that provides new modeling, theoretical, and computational capabilities to cope with data uncertainty in the transportation network context. Following the stochastic programming convention, the distribution parameters can be considered as the first-stage decision and a set of reconstructed demand corresponding to link flow observations in the interested time interval can be treated as second-stage resource variables. As a result, this approach is capable of estimating the population parameters of an underlying probability distribution of the OD demand and simultaneously reconstructing OD trip tables associate with each individual data sample. Besides, the estimation approach provides a great flexibility to accommodate various behavior and operational assumptions and rules as constraints, making it easier to exploit both hard data and soft information. To demonstrate the scalability of the proposed method, we exploit a popular stochastic programming decomposition approach, progressive hedging to address computational challenges due to the problem size and data dimension.

Compared with existing approaches, the major contributions of this study lie in the following:

- From a conceptual perspective, approaching the problem from a fresh stochastic programming (SP) framework, we are able to make connection between the new estimation and the well-studied SP problems. Through analytical and computational results, we demonstrated how the theoretical and computational advancements in stochastic programming can be exploited in the new network estimation problem context.

- From a modeling perspective, we emphasize the importance of having modeling flexibility to incorporate domain knowledge in data-driven estimation processes. As demonstrated from the numerical experiments, by incorporating both network topology information and network operation knowledge in the estimation process, the estimation quality can be largely improved.

The rest of the chapter is structured as follows. We first present a stochastic programming based OD estimation model, accompanied by a consistency analysis. We then implement a

scenario decomposition approach via Progressive Hedging to solve the problem in a scalable manner. Last, we demonstrate the applicability of the proposed network-level demand estimation framework through several numerical case studies, including one highway network and two transit networks.

## 2.2 A Stochastic Programming based OD Demand Estimation Framework

### 2.2.1 Stochastic Programming Estimation Model

In this section, we present a fresh angle of using stochastic optimization techniques for OD demand estimation. Recall a general two stage stochastic nonlinear program (Ruszczyński and Shapiro, 2003)

$$\begin{aligned} \min_{x} \quad & L(x) + E_{\xi}[Q(x,\xi)] \\ \text{s.t.} \quad & g_i^1(x) \leq 0 \\ \text{with} \quad & \\ Q(x,\xi) = \min_{y} \quad & f(y;x,\xi) \\ \text{s.t.} \quad & g_i^2(y;x,\xi) \leq 0 \end{aligned} \tag{2.2.1}$$

Typically, the goal can be interpreted as to make a planning decision that minimizes the total costs including the current and the expected future costs. The first stage planning decision $x$ with cost $L(x)$ has to be made before any possible outcome of the random parameter $\xi$ becomes certain. In the second stage, the actual realization of $\xi$ becomes known and a recourse decision $y$ can be taken. $Q(x,\xi)$ is the objective function of the second stage problem given a particular choice of $x$ and a realization of $\xi$.

In the new context of OD estimation problem, let us approach the demand estimation as a stochastic decision making problem. Consider a transportation network $\mathcal{G}(\mathcal{N},\mathcal{E})$ with $N$ OD pairs and $S$ links, where $\mathcal{N}$ and $\mathcal{E}$ represent the node set and link set, respectively. Consider

that we have $K$ observations of the link flows $v^{(k)} \in \mathbb{R}^S$, with $k = 1, 2, \ldots, K$. An example could be the link flow observations of morning peak period across $K$ days. Our goal is to find a 'good' point estimate for OD demand across $K$ observation intervals. In the meantime, we also want to reconstruct all OD flow for each observation interval $k = 1, 2, \ldots, K$. Thus we can have not only the aggregated statistics to quantify the stochastic OD demand, but also we will have a discrete approximation of the probability distribution of the demand. Adapting to the common stochastic programming language, each link flow observation $v^{(k)}$ can be considered as the results from a particular realization of OD demand $y^{(k)}$ in scenario $k$. Let $x$ be the 'best' estimate of OD demand that we want to estimate following some specific risk measure. The difference between $x$ and the estimated OD demand $y^{(k)}$ in scenario $k$ can be interpreted as a scenario-dependent adjustment/recourse.

The relationship between the $k^{th}$ demand $y^{(k)} \in \mathbb{R}^N$ and its 'best' estimated value $x$ can be expressed as

$$y^{(k)} = x + \pi^{(k)} \tag{2.2.2}$$

where $\pi^{(k)}$ denotes the deviation of the $k^{th}$ scenario from the best estimate of OD flow. The deviation, following the convention of stochastic programming terminology, can be interpreted as a scenario-dependent recourse/adjustment.

Let us express the $k^{th}$ observation of link flow with noise as

$$v^{(k)} = z^{(k)} + \epsilon^{(k)}, \forall k = 1, 2, ..., K \tag{2.2.3}$$

where $z^{(k)}$ is the denoised link flow to be estimated. We assume $\epsilon^{(k)}$ is *iid* zero-mean error across $k$. Note that this assumption does not conflict with the fact that the elements $\epsilon_i^{(k)}$ of each vector $\epsilon^{(k)}$ do not have to be identical across the links ($\forall 1 \leq i \leq S$), and that the magnitude of noise may be dependent on the magnitude of $v^{(k)}$ for each element $i$.

In our design of the loss function, we consider the following three main criteria:

- *Estimated OD flow should align with our prior belief or knowledge of OD demand.* Prior knowledge/belief on the OD demand may be incorporated either in the loss function

as a penalty or in the estimation model as constraints on the deviation from historical observations.

- *Estimated OD flow should produce approximately similar link flow results compared to link flow observations.* For this, we will need to exploit network operation and travel behavior relations to link the OD flow space and link flow space.

- *The estimation process should recognize the uncertain nature of the OD flow and reflect the modeler's risk attitude.* The choice of the risk measures, whether being risk neutral or risk averse, should be a part of the design principle and have a corresponding statistical interpretation.

Following these considerations, and given observations of link flow $v^{(k)}$ $(k = 1, 2, \ldots, K)$ and an unreliable historical prior belief on OD demand, denoted as $w \in \mathbb{R}^N$, we now present our estimation model in a stochastic programming framework. The first stage decision considers the 'best' estimate of demand parameters $x$ following certain risk preference. Prior knowledge on the demand parameters can be incorporated in the first stage via introducing a loss term $l(x; w)$, which is the distortion between the estimated demand parameter $x$ and the prior knowledge input as vector $w$. Besides the prior knowledge of the demand, there are two other places where domain knowledge could be captured:

- *Feasibility sets for the decision variables.* Denote the feasible region of demand $x$ as $\mathcal{F}_1$ and $y^{(k)}$ as $\mathcal{F}_2$. These feasibility sets may include box constraints (such as directly requiring $x$ or $y^{(k)}$ to be within an allowable range), or/and general linear inequality constraints (such as requiring the total trips generated from an origin must be within a certain range). In our numerical experiments in the following section, only non-negativity constraints are included in $\mathcal{F}_1$ and $\mathcal{F}_2$ for illustration purpose.

- *The network assignment mapping, denoted as $G(\cdot)$, that captures the relation between OD demand $y^{(k)}$ and link flow $z^{(k)}$.* Under a normal situation, we do not expect the assignment rules to change from day to day, in which case the mapping $G(\cdot)$ does

not need to be $k$ specific. However, in case there are major disruptions to the system supply side on a certain day $k$, which may impact the network topology, one will need to express the mapping $G^{(k)}(\cdot)$ differently. In the transportation science literature, there are several popular network assignment models. Without loss of generality, in our numerical case studies, we implemented the classic static traffic assignment mapping, expressed as

$$\begin{aligned} &\min_{z} \sum_{e} \int_{0}^{z_e} t_e(a)da \\ &s.t. \sum_{p} f_p^j = y_j, \quad \forall j \\ &z_e = \sum_{p} \sum_{j} f_p^j \delta_{e,p}^j, \quad \forall e \\ &f_p^j \geq 0, \quad \forall j, p \end{aligned} \tag{2.2.4}$$

where $z$ is link flow, $e$ is link index in the network, $j$ is OD pair index, $p$ is path index, $t_e(\cdot)$ is link performance function on link $e$, $f_p^j$ is path flow on path p connecting OD pair $j$. $y_j$ is OD flow on OD pair $j$. $\delta_{e,p}^j = 1$ if link $e$ is on path $p$ connecting OD pair $j$ and $\delta_{e,p}^j = 0$ otherwise.

Following the two stage stochastic programming design, we include the scenario dependent decision and domain knowledge in the second stage problem. Given the first stage decision $x$, the second stage decision for each scenario $k$ is to find reconstructed OD $y^{(k)}$ based on observed link flow $v^{(K)}$, where $y$ takes a discrete empirical distribution with probability $Pr(Y = y^{(k)}) = \frac{1}{K}$. The second stage objective aims for two considerations: 1) to impose the closeness between the first stage variable $x$ and the second stage variable $y^{(k)}$, i.e. to reduce recourse; and 2) to ensure the demand $y^{(k)}$ could produce link flow close to the observations. Then we have the second stage objective of scenario $k$ as

$$Q(y; x, v^{(k)}) = \min_{y^{(k)} \in \mathcal{F}_2} \quad r(x, y) + t(y, v^{(k)}) \tag{2.2.5}$$

By incorporating the assignment mapping $G(\cdot)$, we can translate information on OD demand to link flow, thus the second stage problem can be reduced to

$$\begin{aligned} Q(x, v^{(k)}) = & \min_{y^{(k)} \in \mathcal{F}_2, z^{(k)} \geq 0} \quad r(x, y^{(k)}) + \rho s(z^{(k)}, v^{(k)}) \\ s.t. \quad & z^{(k)} \in G^{(k)}(y^{(k)}) \end{aligned} \tag{2.2.6}$$

The first term in $Q(\cdot)$ measures the scenario-dependent fluctuations, and the second term measures the discrepancy between the observed and estimated link flows. Weighting parameter $\rho$ controls the contribution of the two loss terms. Adopting a risk-neutral preference, plugging in the first stage cost weighted by parameter $\mu$ and the number of scenarios for computational convenience,the two-stage stochastic programming estimation (SPE) can be formulated as

$$\begin{aligned} \mathbf{SPE}(w, v^{(1)}, \ldots, v^{(K)}): \quad & \min_{x \in \mathcal{F}_1} \quad \frac{1}{K} \mu l(x, w) + E[Q(x, v^{(k)})] \\ & \text{with} \\ & Q(x, v^{(k)}) = \min_{y^{(k)} \in \mathcal{F}_2, z^{(k)} \geq 0} \quad r(x, y^{(k)}) + \rho s(z^{(k)}, v^{(k)}) \\ & s.t. \quad z^{(k)} \in G^{(k)}(y^{(k)}), \quad \forall k = 1, \ldots, K \end{aligned} \tag{2.2.7}$$

Functions $l(\cdot)$, $r(\cdot)$, $s(\cdot)$ represent the general risk measures. Here, we adopt a least-squares form. The balance between these error terms is controlled by the weighting factor $\mu$ and $\rho$. The first term in Eq. 2.2.7 is attached with $1/K$ simply for computational convenience. The objective is to minimize the total expected estimation loss while requiring that both the first-stage and second stage decision variables satisfying all feasibility constraints including the network assignment mapping $G^{(k)}$.

It is clear that the above model falls within the classic two-stage stochastic programming framework mentioned above, where $x$ can be considered as the first-stage decision and $y^{(k)}$, $z^{(k)}$ as the second-stage scenario-dependent recourse decision. A deterministic equivalent

program of SPE (2.2.7) then can be expressed as

$$\min_{x\in\mathcal{F}_1, y^{(k)}\in\mathcal{F}_2, z^{(k)}\geq 0} \frac{1}{K}\left(\mu l(x,w) + \sum_{k=1}^{K} r(x,y^{(k)}) + \eta \sum_{k=1}^{K} s(z^{(k)}, v^{(k)})\right) \tag{2.2.8}$$
$$s.t. \quad z^{(k)} \in G^{(k)}(y^{(k)}), \quad \forall k = 1,\ldots,K$$

For the convenience of the readers, below is a summary of key assumptions used for the proposed SPE OD estimation method:

- OD demand is static and evenly distributed within one observation interval $k$, and it repeats over $K$ observation intervals.
- The link flow observation errors are *iid* Gaussian across time horizon $K$.
- The variations of link flow observations results only from variations of OD flows from multiple observations and observation noise. Changes in network supply and configurations are beyond the scope of this study.
- Network structure and cost parameters associated with generalized travel costs are known as input used in the assignment mapping. In addition, travelers are rational and the route choice is based on the generalized travel cost.
- The system analyst adopts a risk neutral attitude for the estimation.

We acknowledge that in reality, it is nearly impossible to expect an assignment model that could perfectly match real-world relation between travel demand and network flows, since motorists are not always rational. In the literature, assignment model specification errors have been considered in studies by Cascetta (1984) and Sherali et al. (1994). In this study, we do not explicitly consider model specification errors, because whether or not to consider such model specification error would not give rise to new challenges in terms of estimation problem structure — model specification errors can be combined with measurement errors and viewed as random fluctuation of flows.

We should also point out that rich stochastic programming literature has demonstrated that different design principles and emphasis can be achieved by plugging in different risk

measures, i.e. how to quantify loss and uncertainties in the objective function and constraints (Eichhorn and Römisch, 2005; Rockafellar, 2018). Some examples include conditional Value-at-Risk (VaR) (Schultz and Tiedemann, 2006), risk averse functional (Noyan, 2012), chance constraints (Kataoka, 1963) and the field of distributional robust optimization (Wiesemann et al., 2014). Even though we only implemented a risk neutral measure, our proposed SPE approach under the stochastic programming framework can flexibly incorporate different risk measures, making it generalizable for various needs in real-world applications. In addition, various estimation principles could fit into the estimation framework, including maximum likelihood estimation from the field of statistics inference and cross entropy based estimation from the field of information theory. These statistical principles have close relationship with risk measures in stochastic programming. In a statistical sense, the 'best' estimate $x$ from our SPE model is equivalent to the mean value of OD demand across $K$ observation sets under a risk neutral preference. The result of $y^{(k)}$ can be interpreted as a discrete set of representative scenarios of OD flow, and they form a discrete probability distribution of OD demand $x$. Furthermore, the sample covariance of the OD could be estimated using reconstructed OD flow $y^{(k)}$ across the entire observation horizon. Note that in Yang et al. (2018), an estimation approach was developed based on a modeling philosophy that integrates the estimation and reconstruction problems, two problems that had been previously studied separately. In this study, we approach the estimation problem from a fresh stochastic programming perspective. It is valuable to note that the two formulations are fundamentally equivalent, even though we approached the problem formulation from a different angle. An advantage of formulating the estimation problem in a stochastic programming framework is that it opens up future opportunities to exploit the rich literature from stochastic programming community in terms of modeling (such as how risks are quantified), analytical (such as proof of statistical properties), and computational (such as decomposition methods) issues, which will be demonstrated in later sections.

### 2.2.2 Consistency Analysis

We want to show that when the number of observations $K$ increase indefinitely, the sequence of estimator for $\hat{x}^K$ from discrete version of SPE converges in probability to $\hat{x}^*$. Such estimate is obtained by solving the following problem where every single possible value of V is considered, as

$$\hat{x}^* = \underset{x\in\mathcal{F}_1}{\operatorname{argmin}} \frac{1}{K}\mu l(x,w) + E_V[Q(x;v)] \quad \text{with} \quad Q(x,v) = \min_{y\in\mathcal{F}_2, z\geq 0} r(x,y) + \rho\tilde{s}(y,v) \tag{2.2.9}$$

where the original $s(\cdot,\cdot)$ and constraints $z^{(k)} \in G^{(k)}(y^{(k)})$ in SPE 2.2.7 are absorbed into one function $\tilde{s}(\cdot,\cdot)$ for compactness. With finite number of scenarios, the expectation of the second stage problem in SPE 2.2.7 can be approximated by the discrete version as

$$\begin{aligned} E_k[Q(x;v^{(k)})] &\approx \sum_{k=1}^{K} Pr(V = v^{(k)}) Q(x, v^{(k)}) \\ &= \frac{1}{K}\sum_{k=1}^{K} Q(x, v^{(k)}) \\ &= \frac{1}{K}\sum_{k=1}^{K} \min_{y^{(k)}\in\mathcal{F}_2, z^{(k)}\geq 0} r(x, y^{(k)}) + \rho\tilde{s}(y^{(k)}; v^{(k)}) \end{aligned} \tag{2.2.10}$$

Let us write the best estimate of $x$ as the solution to SPE from $K$ iid observations as

$$\begin{aligned} \hat{x}^K &= \underset{x\in\mathcal{F}}{\operatorname{argmin}} \frac{1}{K}\mu l(x,w) + \frac{1}{K}\sum_{k=1}^{K} Q(x; v^{(k)}) \\ \text{with} Q(x, v^{(k)}) &= \min_{y^{(k)}\in\mathcal{F}, z^{(k)}\geq 0} \quad r(x, y^{(k)}) + \rho\tilde{s}(y^{(k)}, v^{(k)}) \end{aligned} \tag{2.2.11}$$

We can also write the feasibility constraints into the indicator functions (also named as characteristics function in the field of convex analysis (Rockafellar, 1970)) respectively as

$$I_1(x) = \begin{cases} 0, & x \in \mathcal{F}_1 \\ +\infty, & x \notin \mathcal{F}_1 \end{cases} \tag{2.2.12}$$

$$I_2(y) = \begin{cases} 0, & y \in \mathcal{F}_2 \\ +\infty, & y \notin \mathcal{F}_2 \end{cases} \tag{2.2.13}$$

Then we can rewrite SPE as the minima of the expected value of the following function with respective to $x$

$$\omega(x; v, w) = \mu l(x, w) + Q(x; v) + I_1(x), \text{ with } Q(x; v) = \min_y r(x, y) + \tilde{s}(y; v) + I_2(y) \quad (2.2.14)$$

As a convenient notation, we denote $\omega(x; v^{(k)}, w)$ for each observation set as $\omega^{(k)}$.

Therefore, to prove that $\hat{x}^K \xrightarrow{P} x^*$, we only need to show the problem 2.2.11 converges to 2.2.9 asymptotically. The ideas are built based on the law of large numbers in lower semicontinuous (lsc) functions (Artstein and Wets, 1995; Korf and Wets, 2001).

**Theorem 1.** *If $\omega^{(k)}$ is a sequence of lsc iid functions, the sample mean of $\omega^{(k)}$ will epi-converge to the expected function*

$$\frac{1}{K}\sum_{k=1}^{K} \omega^{(k)} \xrightarrow{epi} E(\omega(x; v, w)) \quad \mu - a.s. \quad (2.2.15)$$

*The minima of the corresponding optimization problem converges in probability*

$$\hat{x}^K = argmin \frac{1}{K}\sum_{k=1}^{K} \omega^{(k)} \xrightarrow{a.s.} x^* = argmin E_V(\omega(x; v, w)) \quad (2.2.16)$$

and this indicates the strong consistency of the SPE.

Proof refer to Theorem 5.2 in Attouch and Wets (1994).

In order to apply the Theorem, we only need to show $\omega$ is a lsc function. By the fact that the sum of real-valued lsc functions is still lsc, we could check the components of $\omega$ individually.

- In the context of convex analysis, the characteristic function of any closed set is lower semicontinuous, and the characteristic function of any open set is upper semicontinuous. For closed sets $\mathcal{F}$, the characteristics functions $I_1(\cdot)$ and $I_2(\cdot)$ are lsc.

- We know l-p norm is continuous, thus it is also lsc function. And our least squares loss functions $l(\cdot, \cdot)$ and $r(\cdot, \cdot)$ fit into this category.

- We have $Q(\cdot,\cdot)$ as the value function of the optimization problem in the second stage. From Theorem 1.17 in Rockafellar and Wets (2009), we know the following two arguments are true: a) the lsc property can be inherited through the minimization, as

$$\tilde{s}(y,v) = s(z,v) \quad with \quad z = g(y) = \underset{z}{\text{argmin}} G(z,y) \tag{2.2.17}$$

  as the result from equilibrium assignment with optimization based $g(y)$ and b) the convex property of function $G(z,y)$ can lead to lsc $g(y)$ as the minima point of $G$ with respect to $z$.

Therefore, with lsc statistical functions and closed constraint sets on both stages, $\omega$ is lsc if g is closed form and lsc, or g is based on a convex program. Thus, we have the estimator of SPE converges to the 'best' point in probability following the proof described above.

## 2.3 A Decomposition Method Using Progressive Hedging

It is apparent that considering many data samples (scenarios) simultaneously could impose computational challenges, as the size of the problem becomes much larger than the counterpart based on a single scenario. Motivated by the effort in large scale stochastic programming literature (Carøe and Schultz, 1999; Collado et al., 2012), we leverage a scenario decomposition approach to cope with the scalability issue for our demand estimation problem. Among the decomposition methods, progressive hedging (PH) is a scenario-based decomposition technique that can be used to solve large scale stochastic programs (Rockafellar and Wets, 1991). The scenario decomposition is performed by relaxing the non-anticipativity constraints, allowing solving each scenario sub-problem independently. More specifically, the non-anticipativity constraints are included in the revised objective function as penalty along with multiplier terms, and are progressively enforced by an iterative procedure. For each scenario sub-problem, we obtain approximate solutions for the problem of minimizing the

deterministic component plus terms that penalize lack of implementability, subject to the constraints. Algorithm 1 summarizes the PH algorithm for the SPE problem.

**Algorithm 1:** Progressive hedging for SPE OD estimation

**Data:** Data $w$, $v^{(k)}$, $\forall k = 1, 2, \ldots, K$, penalty weight $r$, loss weighting parameter $\mu$ and $\eta$, tolerance $\epsilon$

**Initialize:** *Vector* $\hat{x}^0$*, vector* $\rho_0$*,* $G^{(k)}$ *computed using historical OD flow* $w$

**for** *Iteration index* $i = 1$ *to max iteration* **do**

  **for** *Scenario index* $k = 1, \ldots, K$ **do**

    Let $(x^{(k),i+1}, y^{(k),i+1})$ solve the quadratic program for $k$th subproblem

$$\min_{(x^{(k)}, y^{(k)})} \frac{1}{K} \Big( \mu \|x^{(k)} - w\|_2^2 + \|x^{(k)} - y^{(k)}\|_2^2 + \eta \|z^{(k)} - v^{(k)}\|_2^2 + \rho^{(k),i^\top} x^{(k)} + \frac{r}{2} \|x^{(k)} - \hat{x}^i\|_2^2 \Big)$$

$$\text{s.t.} \quad z^{(k)} = G^{(k)}(y^{(k)})$$

$$x^{(k)} \in \mathcal{F}_1$$

$$y^{(k)} \in \mathcal{F}_2$$

  **end**

  Update $G^{(k)}$ using newly estimated $y^{(k)}$ for all $k = 1, \ldots, K$

  From the solution for K subproblems, compute $\hat{x}^{i+1} = \sum_{k=1}^{K} Pr(k)(x^{(k),i+1})$

  Update dual variable estimates $\rho^{(k),i+1} = \rho^{(k),i} + r(x^{(k),i+1} - \hat{x}^{i+1})$

  Terminate if $\|x^{(k),i+1} - x^{(k),i}\| < \epsilon$ and $\|\rho^{i+1} - \rho^i\| < \epsilon$, otherwise continue

**end**

[1]

Significant computation time can be saved using these parallelizable smaller-scale subproblems, as will be demonstrated later in the numerical section. Another benefit of scenario-decomposition approaches comes to its equivalent manner as sequential learning (Shalev-Shwartz et al., 2012) – since we can decompose the master problem SPE 2.2.7 into subproblems for each observation as scenario, we can sequentially update the estimates in an online fashion whenever a new batch of observations arrives. This feature is out of the scope of the

[1] In our numerical examples, the assignment matrix did not vary after one iteration, therefore we fixed $G^{(k)}$ after one iteration in the PH algorithm to speed up the process. There are other speedup procedures reported in the PH literature, including fixing certain elements of the unknown vectors if they do not change after some iterations.

current study, thus will not be exploited in this study.

PH is based on the proximal point method. Theoretically, it is proven to converge to a global optimal solution for convex problems (Rockafellar and Wets, 1991), and a local optimal for a nonconvex problem. Numerically, it is known that PH is sensitive to the choice of the penalty parameter, in our case $r$ in the subproblem. The penalty parameter weighs the penalty term in the augmented Lagrangian function and specifies the step size in the dual variable updates. There is not yet theoretical support of the best choice of the penalty parameter. In practice, dynamic adjustment of the parameter across iterations turned out to be beneficial (Mulvey and Vladimirou, 1991; Hvattum and Løkketangen, 2009; Zehtabian and Bastin, 2016), which was implemented in our case studies.

## 2.4 Numerical Case Studies

In this section, we present three case studies on OD estimation for one highway network and two transit networks to demonstrate the applicability and effectiveness of our proposed SPE method.

### 2.4.1 OD Estimation for Highway Network

We begin with a city-scale highway network of Berlin Friedrichshain (Transportation Networks for Research Core Team, 2021). The highway network consists of 23 zones, 529 OD pairs, 523 links and 224 nodes, as shown in Figure 2.1. In our case study, we use equilibrium-based traffic assignment to simulate traffic flows in the network and use our SPE method to estimate the unknown demand, assuming we only have an unreliable historical demand vector and $K$ observations of link flows. The synthetic true OD demand is assumed to follow a multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$. We allow a block diagonal covariance structure, meaning the OD pairs are more correlated with themselves and nearby locations compared to locations farther away. With given synthetic OD flow, the observed link flow is obtained by running congested User Equilibrium traffic assignment, following the BPR link performance

functions. The $K = 400$ counts of OD demand, covariance matrix and observed link flows are shown in Figure 2.2. Each column in the heatmap 'Y true' and 'V observed' represents a data column vector, totaling $K = 400$ columns. Each row in 'Y true' represents one OD pair, and each row in 'V observed' represents one link in the network.

In our experiments, we varied the number of unknown OD pairs $P \leq N$, to mimic different problem sizes. Consider $P$ unknown OD pairs and full knowledge of the rest $N - P$ OD pairs. For each choice of $P$, we randomly select without replacement $P$ OD pairs that are unknown. The process is repeated 10 times.

Figure 2.3 reports the box plot of the mean absolute percentage error (MAPE) of the mean estimate as $P$ varies. From the results, we can see that the quality of the mean estimates is consistently good across varying $P$ with sufficient number of scenarios ($K = 600$ in this case). On the other hand, the error of the reconstructed scenario-dependent demand increases while the number of unknown increases, as shown in Figure 2.4.

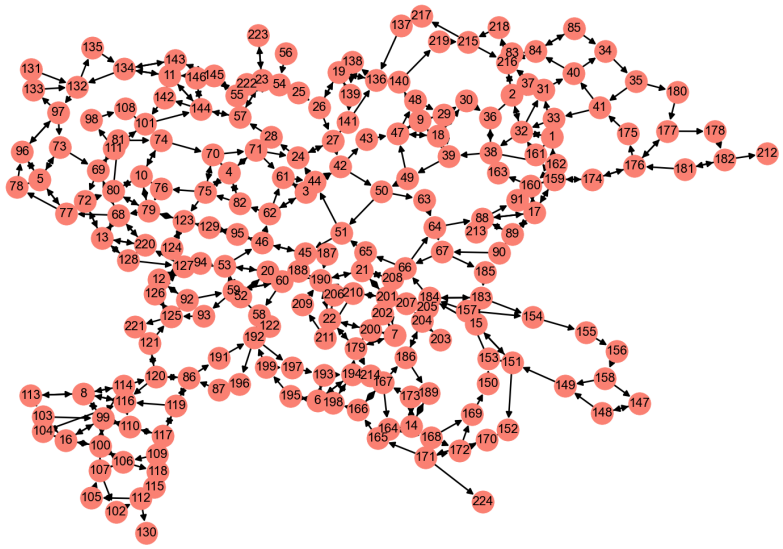


Figure 2.1: Berlin-Friedrichshain network

The PH decomposition method has shown promising improvement in computational time

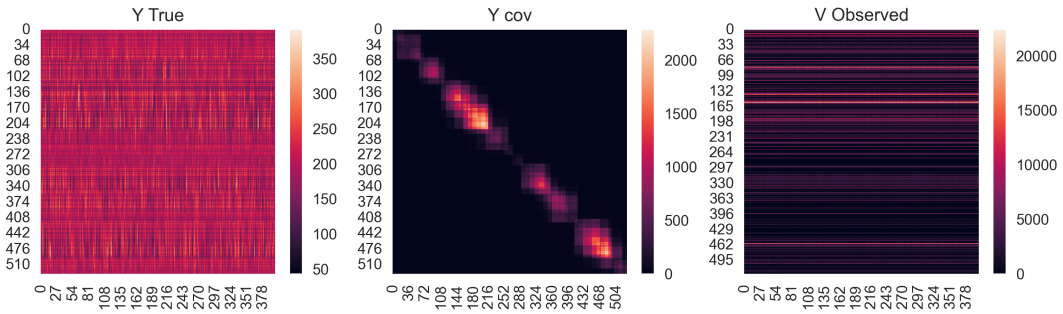



Figure 2.2: True OD demand, covariance of OD demand, and observed link flow

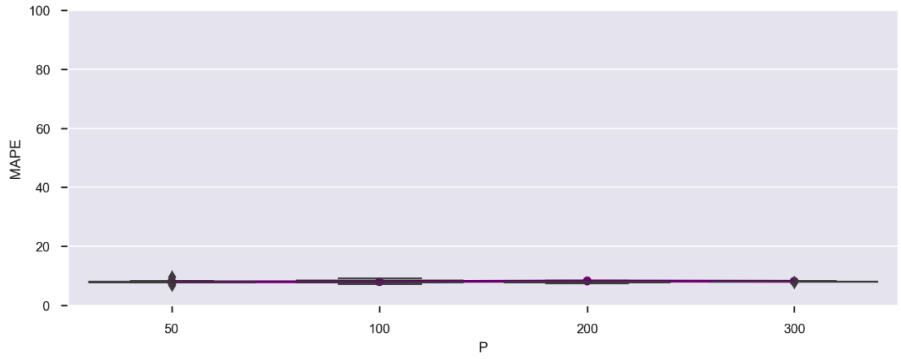



Figure 2.3: MAPE for mean estimates $x$ for experiments of P

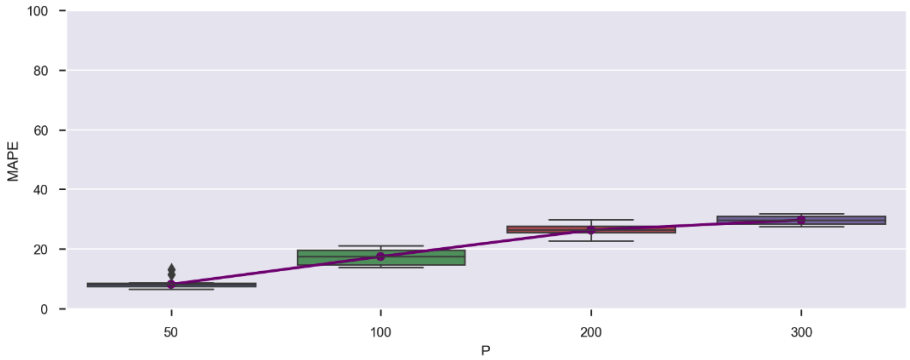



Figure 2.4: MAPE for all K = 600 OD flow estimates $y^{(k)}$ for experiments of P

over standard solvers. For example, on a MacBook pro 2019 with 16 GB memory, the machine runtime ranges from 10 seconds to 2.4 hours for problem sizes with P from 50 to 400 and K from 200 to 400. As a comparison, a standard solver CVXOPT library[2] took more than 2.5 hours to converge for the problem with $K = 50$ and $P = 500$. When K is increased to 200, out-of-memory issue occurred as a large number of large-scale demand vectors to be estimated all at once.

### 2.4.2 OD Estimation for Transit Network

Similar as for highway networks, understanding OD demand is also important for transit systems, since network-level transit demand serves as a critical input in transit planning and operation decision processes. Traditionally, transit operators heavily relied on costly passenger travel surveys for demand forecasting and transit system planning, which could potentially involve relatively high level of bias (Cascetta, 1984). More recently, advanced automatic passenger counting (APC) systems are brought into transit operators to help understand passengers' travel demand. APC provides a full spatial and temporal coverage of passenger counts across the system compared to other data sources, where the extracted information can be utilized to estimate OD matrices. Different from the highway network case, there are transfer options in transit systems that would involve more complicated network modeling and route choice behavior. In this section, we implement the proposed OD estimation model in two transit networks to explore its applicability in the transit context, which tends to have different network structures and behavior assumptions from the highway situation.

Transit OD flow estimation problem based on passenger counts can be categorized into two categories: route-level OD estimation (Ben-Akiva et al., 1985; McCord et al., 2010; Hazelton, 2010; Ji et al., 2014) and network-level OD estimation (Nguyen et al., 1988; Wong and Tong, 1998; Wong et al., 2005). The case studies reported here focus on network-level OD estimation. In this context, the passenger counts are analogous to the aggregated link

[2]https://cvxopt.org/

flows in a general road network. Since the APC data are collected at a vehicle level, which is not directly equivalent to link-level passenger flow in the network, we had to first conduct a data transformation to accommodate the difference in data collection as described in the following section.

#### 2.4.2.1 Transit and automatic passenger counting data

A transit line is determined by a sequence of stops, and it can be further described using the frequency of transit vehicles and vehicle types. Passengers can choose to board and alight at transit stops along the transit line. Figure 2.5 illustrates a toy transit system with four transit lines and four transit stops. The transit network can be described as a directed graph $G_{LS}(\mathcal{N}, LS_{\mathcal{E}})$. The node set $\mathcal{N}$ represents transit stops where passengers can board and alight, and the link set $LS_{\mathcal{E}}$ represents transit line segments. A line segment $LS_{e_{(i,j)}}$, with $i, j \in \mathcal{N}$ is defined as the directed link between two consecutive transit stops $i$ and $j$ along a specific transit line. The transit lines and stops can be converted to collections of links in the network representation, as shown in Figure 2.6. In the example network, transit line 1 serves stop $n_1 \rightarrow n_4$, as an express transit service using dedicated infrastructure. Transit line 2 serves node $n_1 \rightarrow n_2 \rightarrow n_3$, transit line 3 serves node $n_2 \rightarrow n_3 \rightarrow n_4$ and transit line 4 serves node $n_3 \rightarrow n_4$. Transit line 2,3, and 4 share the same roadway infrastructure. Notice that traveling in a transit network usually consists of several steps: 1) arrival at a transit stop; 2) waiting for the bus to get on board; 3) traveling on the bus; 4) transfer when necessary; and 5) arrive at the destination stop. Thus the one stop node $n_i \in \mathcal{N}$ can be decomposed into arriving/alighting node (denoted as $n_i^a$), waiting links and boarding node to line $j$ (denoted as $n_{ij}^b$) associated with each transit line, shown in Figure 2.6. For example, consider a passenger who wants to travel from stop $n_1$ to stop $n_3$. She would first arrive at the transit stop $n_1^a$ (arrival node) and wait for the arrival of the bus in waiting link $(n_1^a, n_{12}^b)$. After she gets on board the bus at node $n_{12}^b$ (boarding node), she would travel in-vehicle through link $(n_{12}^b, n_2^a)$, $(n_2^a, n_{21}^b)$, $(n_{21}^b, n_3^a)$ using transit line L2 and then get off

the bus at the destination $n_3$. She has two options at node $n_2^a$, staying in transit line L2 through link $(n_2^a, n_{22}^b)$ or transferring to line L3 through link $(n_2^a, n_{23}^b)$. Since we assume all passengers are rational, meaning no one would alight and board at the same stop for the same transit line, the second option of transferring can be omitted. Here, the no-transfer behavior at stop $n_2$ along transit line L2 can be implemented by assigning zero waiting time for waiting link $(n_2^a, n_{22}^b)$. The transfer behavior can then be presented by assigning positive waiting time instead. Note there is a one-to-one relationship between a waiting link and the immediate next traveling line segment link. Thus, for simplicity, the network representation with waiting links (shown in Figure 2.6) can collapse into a simpler representation (shown in Figure 2.8). Another complication is due to the common line problem where some parallel transit lines serve the same group of links and stops, a detailed discussion and treatment on network transformation is included in the next section. All the transit experiments below are conducted based on transformed transit network using route-section representation.

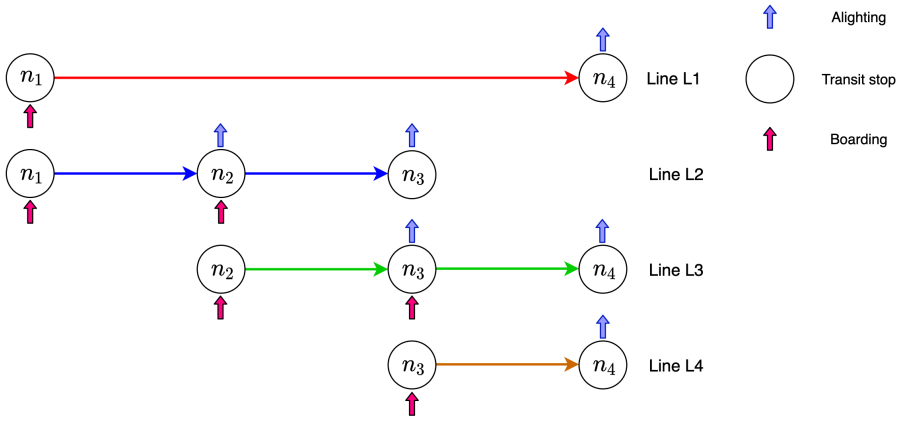



Figure 2.5: Example transit system

APC, denoted as the set $\{I, O\}$, records the numbers of passengers' boarding $I_{n_i}$ and alighting $O_{n_i}$ at transit stop $n_i$. We assume passengers are rational,in the sense that 1) no passenger alights at the first stop, or 2) boards at the last stop or 3) boards and then alights at the exact same stop. For one bus run $u$ from the starting to the end terminals, the line segment link flow $v_{LS_e}$ for a line segment $LS_{e_{(m-1,m)}} \in LS_{\mathcal{E}}$ connecting consecutive stops

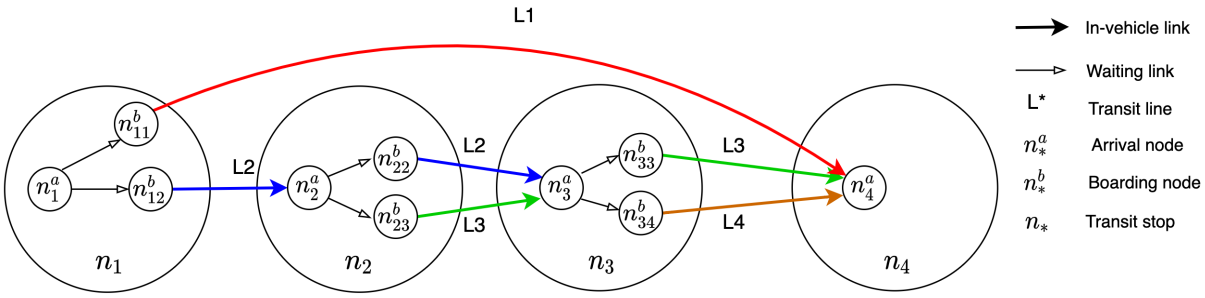



Figure 2.6: Example transit network with waiting links

$m-1$ and $m$ can be obtained by accumulating all the boarding counts and subtracting all the alighting counts along this transit line until the interested stop $n_m$,

$$v_{LS_e}^u = \begin{cases} \sum_{i=1}^{m-1}(I_{n_i}^u - O_{n_i}^u) & m > 1 \\ I_{n_1}^u & m = 1 \end{cases} \tag{2.4.1}$$

where $i_n$ is the boarding counts at stop $n$, $o_n$ is the alighting counts at stop $n$, $m$ is the tail node (downstream stop) of interested segment $LS_{e_{(m-1,m)}}$, $u$ is a specific transit bus run (bus trip) with total runs $U$ in the interested time interval. Consider there may be multiple bus runs in the interested time interval, we have aggregated line segment link flow $v_{LS_e}$ as

$$v_{LS_e} = \sum_{u \in U} v_{LS_e}^u \tag{2.4.2}$$

where $u$ is a specific bus run out of total number of bus runs $U$. Through tracing the number of boarding and alighting along the bus runs for all transit routes, we can have the line segment link flow for the entire network.

#### 2.4.2.2 Frequency-based transit network construction and transformation

Often times, there exist several transit lines running parallel with some stops in common. In the network example shown in Figure 2.8, sections between node 2 and node 3 and sections between node 3 and node 4 are both served by two different transit lines. Chriqui and Robillard (1975) referred this phenomenon as the common line problem: there exist some common sections shared by multiple transit lines and passengers need to select the bus (belong to

certain transit lines) they want to take. From the network modeling perspective, having only one directed link connecting two consecutive nodes would make network representation and transit assignment analysis simpler. Therefore, we adopt the concept of network transformation previously suggested in De Cea and Fernández (1993), and we designed a computer algorithm to fully automate this transformation process.

Let us use a four-node directed network (See Figure 2.7), which was originally presented in De Cea and Fernández (1993), as an example to illustrate the transformation process. Denote the transformed route section network as $\tilde{G}(\mathcal{N}, RS_{\mathcal{A}})$, with $N$ as the node set (same as the node set $\mathcal{N}$ in the original network $G_{LS}(\mathcal{N}, LS_{\mathcal{E}})$) and $RS_{\mathcal{A}}$ as the set for route section links. Let $LS_e \in LS_{\mathcal{E}}$ be a line segment for original network $G_{LS}(\mathcal{N}, LS_{\mathcal{E}})$, $RS_a \in RS_{\mathcal{A}}$ as a route section in modified network $\tilde{G}(\mathcal{N}, RS_{\mathcal{A}})$ and $Li$ as a transit line $i$. The original network includes 6 line segment links $LS_1, LS_2 \ldots LS_6$ and 6 OD pairs $(n_1 \rightarrow n_4)$, $(n_1 \rightarrow n_3)$, $(n_1 \rightarrow n_2)$, $(n_2 \rightarrow n_4)$, $(n_2 \rightarrow n_3)$ and $(n_3 \rightarrow n_4)$. It is served by 4 transit lines $L1, L2, L3, L4$, as described in the previous section. The route section is defined as a portion of a transit line between two consecutive transfer/boarding-alighting stops. Note one route section can correspond to multiple transit lines. In the example, route section $RS_1$ from $n_1$ to $n_4$ is served by line $L1$. Route section $RS_2$ from $n_1$ to $n_3$ is served by line $L2$. Route section $RS_3$ from $n_1$ to $n_2$ is served by line $L2$. Route section $RS_4$ from node $n_2$ to $n_3$ is served jointly by $L2$ and $L3$. Route section $RS_5$ from node $n_3$ to $n_4$ is served jointly by $L3$ and $L4$. Route section $RS_6$ from node $n_2$ to $n_4$ is served by $L3$. Following the logic of route-section representation, this transit network can be transformed into a modified network with 6 route-section links, where the shortest paths (including transfers) connecting each OD pairs can be easily identified and computed.

Now let us construct the relation of passenger flows between line segments $LS_{\mathcal{E}}$ and route sections $RS_{\mathcal{A}}$ in the network transformation. Note that some route sections are served by multiple lines, such as route section $n_2 \rightarrow n_3$ (served by line $L2$ and $L3$), and route section $n_3 \rightarrow n_4$ (served by line $L3$ and $L4$) respectively. It is reasonable to assume that a transit

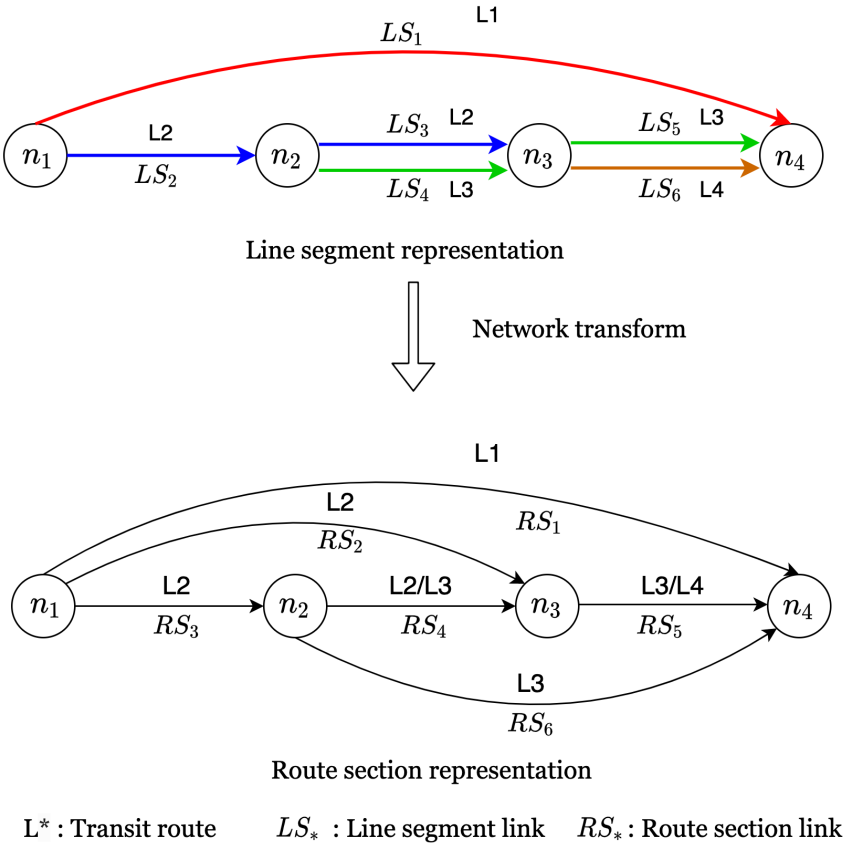



Figure 2.7: Transit network transformation

line with higher frequency takes higher proportion of aggregated passenger flow of the route section, expressed as

$$v_{LS_e} = \sum_{U_{RS_a} \in \mathcal{S}_e} \frac{f_{LS_e}}{\sum_{LS_i \in \mathcal{S}_a} f_{LS_i}} u_{RS_a}, \tag{2.4.3}$$

where $v_{LS_e}$ represents the passenger flow of line segment link $LS_e$ (Upper half of Figure2.7). $u_{RS_a}$ represents the aggregated passenger flow of the corresponding route section link $RS_a$ (lower half of Figure 2.7) and $\mathcal{S}_e$ represents the set of transit route sections corresponds to route line segment $LS_e$. $\mathcal{S}_a$ denote the set of transit frequencies of transit lines included in route section $RS_a$. $f_{LS_e}$ represents the frequency of transit line serving line segment $LS_e$.

In the example, for line segment link $LS_1$ in transit route $L1$, the flow $LS_{v_1}$ is preserved

to route section $RS_1$, as route section flow $RS_{u1}$, thus line segment flow equals route section flow, $v_{LS_1} = u_{RS_1}$. The flow passing through the line segment link $LS_2$ using transit route $L2$ can be decomposed into: traveling from stop $n_1$ to stop $n_3$ through route $L2$ and from stop $n_1$ to stop $n_2$ through route $L2$ (in the route section representation), thus $v_{LS_2} = u_{RS_2} + u_{RS_3}$. For line segment link $LS_3$, it includes: the passenger flow traveling from stop $n_1$ to stop $n_3$ using route $L2$ (since the flow traverses both $LS_2$ and $LS_3$) and the flow traveling from stop $n_2$ to stop $n_3$ using route $L2$ (as the blue arrows in the upper half of Figure 2.7). Since the route section $RS_4$ is shared by transit routes $L2$ and $L3$ (blue and green arrows in the upper half of Figure 2.7), the line segment link flow is re-distributed based on the relative frequency of these two transit routes, as $v_{LS_3} = u_{RS_2} + \frac{f_{LS_2}}{f_{LS_2}+f_{LS_3}} u_{RS_4}$. Similarly, we have line segment flows as $v_{LS_4} = \frac{f_{LS_3}}{f_{LS_2}+f_{LS_3}} u_{RS_4} + u_{RS_6}$, $v_{LS_5} = \frac{f_{LS_3}}{f_{LS_3}+f_{LS_4}} u_{RS_5} + u_{RS_6}$ and $v_{LS_6} = \frac{f_{LS_4}}{f_{LS_3}+f_{LS_4}} u_{RS_5}$. Follow the transformation described, we obtain a system of equations between link flows of the two different network representations in a vector-matrix form

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{pmatrix}_{S\times 1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{f_2}{f_2+f_3} & 0 & 0 \\ 0 & 0 & 0 & \frac{f_3}{f_2+f_3} & 0 & 1 \\ 0 & 0 & 0 & 0 & \frac{f_3}{f_3+f_4} & 1 \\ 0 & 0 & 0 & 0 & \frac{f_4}{f_3+f_4} & 0 \end{pmatrix}_{S\times L} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{pmatrix}_{L\times 1} \tag{2.4.4}$$

Thus the topological relationship between the original transit network $G_{LS}(\mathcal{N}, LS_{\mathcal{E}})$ and the transformed network $\tilde{G}(\mathcal{N}, RS_{\mathcal{A}})$ can be written compactly as

$$v = Mu, \tag{2.4.5}$$

where link flow vector of the original line segment based network is $v \in \mathbb{R}^S$ and link flow vector of the modified route section based network is $u \in \mathbb{R}^L$. Note that line segment flow vector $v$ can be calculated based on APC data, as described in the next section. Usually we have $S \leq L$, since auxiliary links are added to the modified network during the transformation

process, and the transformation matrix $M \in \mathbb{R}^{S \times L}$ is full row rank. We designed an algorithm to calculate the transformation matrix $M$, as shown in Algorithm 2. This automated procedure helps prepare a route-representation network ready for the transit assignment model.

**Algorithm 2:** Computing Network Transformation $M$

**Data:** $L_{RS}$: A 2D matrix of route section set representing the transit network, with column index as route id, and row index as route section index in the route. $L_{RS}[i][j]$ represents the route section $i$ on the transit route $j$.

**Initialize:** *Transformation matrix* $M \leftarrow$ *all zero array of dimension* $S \times L$

**for** $j$ = *1 to* $L$ **do**

  $L_{temp} \leftarrow$ empty list

  $s \leftarrow$ source node of link $j$

  $t \leftarrow$ target node of link $j$

  Assign transit routes served by route section j to list R, $R \leftarrow L_{RS}[j][:]$ `// R is the list consists of all routes passing through route section j` **if** $length(R) = 1$ **then**

    $L_{selected} \leftarrow L_{RS}[:][routeid = R.id]$

    Search link id $L_{lb} \in L_{selected}$ whose source node $= s$

    Search link id $L_{ub} \in L_{selected}$ whose target node $= t$

    $L_{temp} \leftarrow$ integer sequence from $L_{lb}$ to $L_{ub}$

    **for** $i$ *in* $L_{temp}$ **do**

      $M[i][j] \leftarrow 1$

    **end**

  **else**

    Calculate cumulative frequency of link j, $F = \sum f_k$, $\forall k \in R$

    **for** $k$ *in* $R.id$ **do**

      $L_{temp} \leftarrow$ empty list

      $L_{selected} \leftarrow L_{RS}[:][routeid = k]$

      Search link id $L_{lb} \in L_{selected}$ whose source node $= s$

      Search link id $L_{ub} \in L_{selected}$ whose target node $= t$

      $L_{temp} \leftarrow$ integer sequence from $L_{lb}$ to $L_{ub}$

      **for** $i$ *in* $L_{temp}$ **do**

        $M[i][j] \leftarrow \frac{f_k}{F}$

      **end**

    **end**

  **end**

**end**

#### 2.4.2.3 Experiments based on a toy transit network

The first experiment of transit OD estimation is based on the toy transit example, as shown in Figure 2.8. Different from a road network, a transit network consists of transit lines (transit routes) and transit stops. The APC provides passenger flow observations across the transit system. The APC dataset was synthesized with $K = 250$ assuming the true OD demands follow independent normal distributions. Table 2.1 lists the 6 OD pairs and 11 possible paths of the route-representation transit network. The observation errors of passenger counts follow a set of independent and identically distributed (iid) normal distribution $\mathcal{N}(0, 2^2)$ with mean value to be 0 and variance to be 4. The route choice parameter in the MNL model $\beta$ is set to be 0.8, and the weighting parameter $\rho$ of the cost function is 1. The link travel times are $[10, 6, 11, 5, 10, 4.45]$ minutes for the six line segments. Other operation related parameters related to the network are given in Table 2.2.

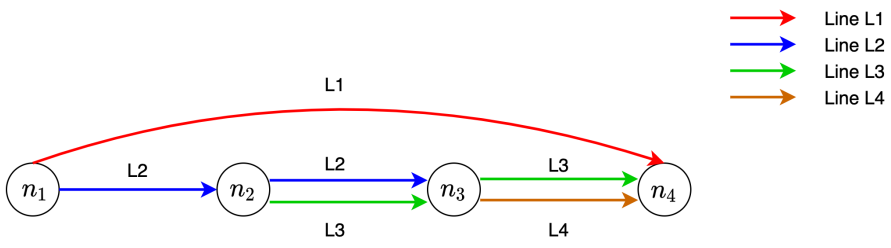



Figure 2.8: Example transit network

Three synthetic examples are generated using different standard deviation values of the true OD demand ($\sigma_i = \sqrt{\mu_i}/5$, $\sigma_i = \sqrt{\mu_i}/10$, $\sigma_i = \sqrt{\mu_i}/20$). Figure 2.9 compares the density plots of the true OD demand (solid curve) and the reconstructed demand (dashed curve[3]) for each OD pair. The RMSE of reconstructed OD demands is $0.77, 0.47, 0.42$ for the three testing cases, respectively. The true mean demands and the estimated mean values of the demands, $\theta$, are reported in Table 2.3. These results demonstrate the estimation quality when

[3]For clean visualization purpose, we used the 'kernel density estimation' feature from python density plot function 'seaborn.distplot' (where the function fits Gaussian kernel density estimate to make it look continuous), alternatively it would show a discrete set of bars for the histogram.

Table 2.1: OD and path for the small transit network

| Path No. | OD No. | From | To | Path |
|---|---|---|---|---|
| 1 | 1 | 1 | 4 | [1, 4] |
| 2 | 1 | 1 | 4 | [1, 2, 4] |
| 3 | 1 | 1 | 4 | [1, 3, 4] |
| 4 | 1 | 1 | 4 | [1, 2, 3, 4] |
| 5 | 2 | 1 | 3 | [1, 3] |
| 6 | 2 | 1 | 3 | [1, 2, 3] |
| 7 | 3 | 1 | 2 | [1, 2] |
| 8 | 4 | 2 | 4 | [2, 4] |
| 9 | 4 | 2 | 4 | [2, 3, 4] |
| 10 | 5 | 2 | 3 | [2, 3] |
| 11 | 6 | 3 | 4 | [3, 4] |

Table 2.2: Operation information of the small transit system

| Route No. | Headway (min) | Stop |
|---|---|---|
| 1 | 8 | 1-4 |
| 2 | 6 | 1-2-3 |
| 3 | 6 | 2-3-4 |
| 4 | 5 | 3-4 |

information of both network topology and operations, as transportation domain knowledge, is taken into account in the SPE demand estimation model.

Next, based on t-statistic, we construct confidence intervals of the difference between two population means, as $\mu_1$ from true OD flow used for the synthetic example and $\mu_2$ from reconstructed OD flow by solving the OD estimation SPE. Here we assume the data samples from these two populations are normally distributed and the variances of the two independent population groups are equal. We also assume independence for OD flows among 6 OD pairs. Denote $\mu_1^i, \forall i = 1, 2, \ldots, 6$ and $\mu_2^i, \forall i = 1, 2, \ldots, 6$ as the population mean of true OD flow and reconstructed for OD pair $i$, respectively. The confidence intervals of $\mu_1^i - \mu_2^i$,

with $\alpha = 0.05$, are shown in Table 2.4 for the 6 OD pairs. Each confidence interval may be interpreted as that we are 95% confident that the true value of $\mu_1^i - \mu_2^i$ falls within the interval. One can see that most of the confidence intervals reported in the table have a small range that covers 0 value, meaning $\mu_1^i$ and $\mu_2^i$ are close to each other within an acceptable tolerance. The only exception is OD pair 5 with variance 20, though the interval is still very close to 0. These consistent results demonstrate confidence in our estimation results.

Table 2.3: Estimation comparison of mean demand $\theta$

| OD Pair | True Mean | Est Mean $\theta$ (Var=5) | Est mean $\theta$ (Var=10) | Est mean $\theta$ (Var=20) |
|---|---|---|---|---|
| OD 1 | 150.00 | 150.10 | 150.05 | 150.02 |
| OD 2 | 120.00 | 120.01 | 119.98 | 119.96 |
| OD 3 | 100.00 | 100.01 | 100.05 | 100.07 |
| OD 4 | 80.00 | 80.05 | 80.05 | 80.05 |
| OD 5 | 60.00 | 60.09 | 60.08 | 60.07 |
| OD 6 | 50.00 | 49.95 | 49.99 | 50.01 |

Table 2.4: Confidence intervals of the difference between the true and the estimated mean values of the demand

| OD Pair | C.I. (Var=5) | C.I. (Var=10) | C.I. (Var=20) |
|---|---|---|---|
| OD 1 | (-0.42, 0.36) | (-0.26, 0.19) | (-0.19, 0.13) |
| OD 2 | (-0.53, 0.16) | (-0.38, 0.01) | (-0.32, -0.05) |
| OD 3 | (-0.30, 0.31) | (-0.17, 0.18) | (-0.12, 0.13) |
| OD 4 | (-0.38, 0.14) | (-0.27, 0.04) | (-0.24, 0.00) |
| OD 5 | (-0.13, 0.34) | (-0.03, 0.24) | (0.01, 0.20) |
| OD 6 | (-0.16, 0.26) | (-0.09, 0.18) | (-0.06, 0.15) |

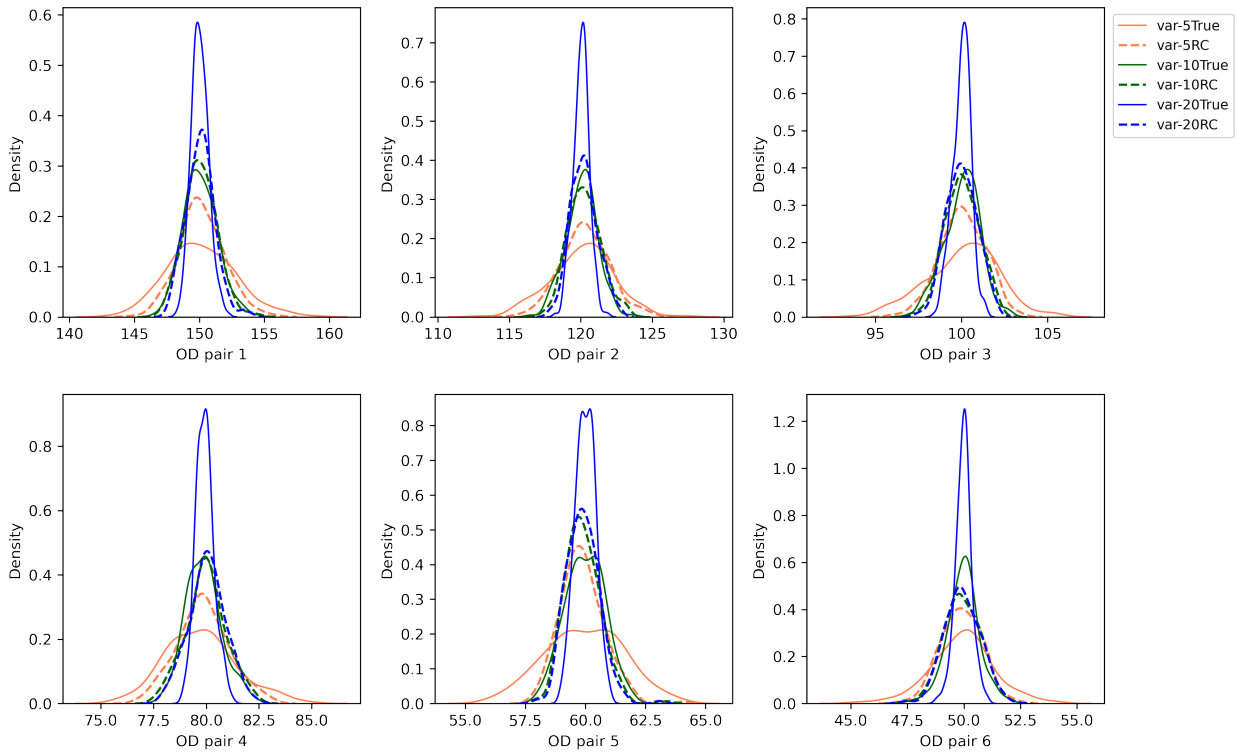



Figure 2.9: Comparison between true demand and reconstructed demand incorporating both topology and transit network operations information

#### 2.4.2.4 An example of real-world scale

The second numerical example for transit is based on Bay Area Rapid Transit (BART)[4], a real-world transit system in San Francisco Bay Area in California, USA. Note that even though the BART system does not use APC based counting system, it serves an ideal test case for our study because of the rare availability of the ground truth demand data collected from Clipper cards. The BART network configuration is shown in Figure 2.10. The operation information (route, travel time and frequency) was obtained from General Transit Feed Specification (GTFS) open source data based on 2019 schedules. For a typical weekday, BART network consists of 48 transit stops/terminals and 7 transit lines. There are 181 directed links in the line segment representation and we obtained 1464 route section

[4]https://www.bart.gov/

links after network transformation using the algorithm we developed 2. Each stop serves as an origin and a destination, respectively. Since no trip in the transit system starts and ends at the same station, we have $48 \times (48 - 1) = 2256$ OD pairs in the network. For each OD pair, up to four shortest paths were computed. Two OD pairs turned out to have only a single path for each. Totally there are 9018 paths considered in the network.

In the synthesized example, we assume the true demands for all the OD pairs across the network follow a set of independent normal distributions $\mathcal{N}_i(\mu_i, \sigma_i), \forall i \leq D$. The mean demand parameters $\mu_i$ follow the ground truth of BART system collected from on and off tapping of the Clipper cards. For each OD pair, we tested four different standard deviation settings, including $\sigma_i = 0$, $\sigma_i = \mu_i/3$, $\sigma_i = \mu_i/5$, $\sigma_i = \mu_i/10$, $\sigma_i = \mu_i/20$. APC observations are synthesized for $K = 250$ observation time periods. Weighting parameter $\rho$ is set to 1. Box plots of the difference between the true and the reconstructed OD flows are given in Figure 2.11, and the comparisons for the true mean values and estimated values are plotted in Figure 2.12. These results indicate reasonably good estimation quality.

Table 2.5: Estimation error for different numbers of scenarios K and standard deviation $\sigma_i$

| | Std. | $\sigma_i = 0$ | $\sigma_i = \mu_i/20$ | $\sigma_i = \mu_i/10$ | $\sigma_i = \mu_i/5$ | $\sigma_i = \mu_i/3$ |
|---|---|---|---|---|---|---|
| RMSE | K = 100 | 216.14 | 229.08 | 237.17 | 251.58 | 275.44 |
| | K = 250 | 237.33 | 242.50 | 242.42 | 231.57 | 260.46 |
| | K = 500 | 228.52 | 226.96 | 220.56 | 222.18 | 260.77 |

To study the impact on the sample size and data variation, Table 2.5 includes the estimation error under different combinations of sample size K and standard deviation $\sigma_i$ for BART network. Here we pick root mean squared error (RMSE) metric to compare among the different settings, since it is more sensitive to variations and at the same scale of the original data. It is clear that RMSE increases as the standard deviation increases in the synthetic demand data $x^{(k)}$. This is because larger variation in $x^{(k)}$ would lead to more uncertainties in observed passenger flow $z^{(k)}$ via transit assignment step $G(\cdot)$, thus causing

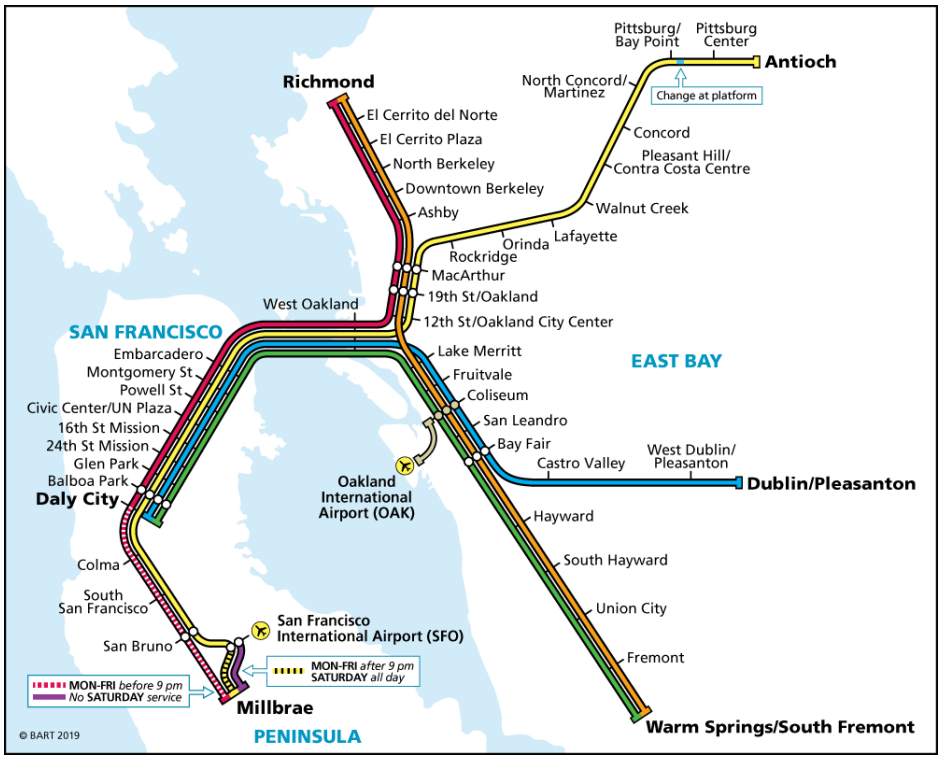



Figure 2.10: BART system map in year 2018 (Source: `https://www.bart.gov/system-map`)

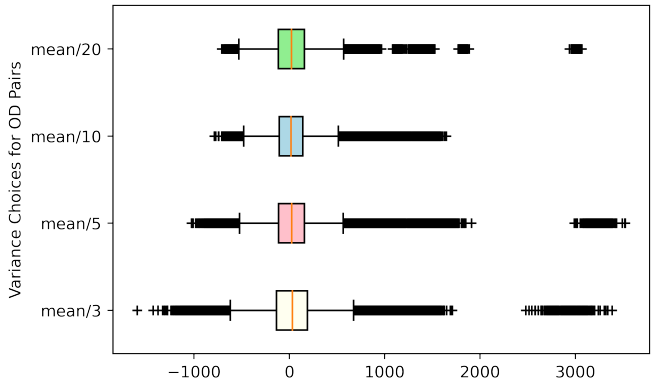



Figure 2.11: Comparison between true and reconstructed demand

variation in the demand reconstruction process in our estimation framework. As for the impact of K on RMSEs, we have observed some improvement in RMSEs when the sample size increased from 250 to 500 in most cases, but the result is too limited to be generalized, considering the sample size of 500 is relatively small for the problem of such dimension.

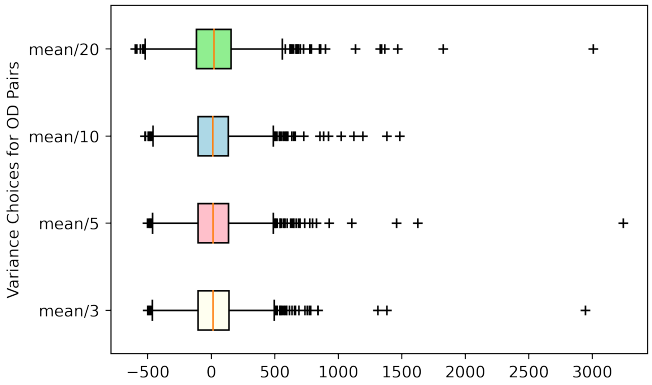



Figure 2.12: Comparison between true mean demand and estimated mean demand

Yet the promising results from these numerical experiments in terms of estimation quality and computational efficiency demonstrated potential applicability of the proposed estimation framework in real-world problems.

## 2.5 Conclusions and Discussions

In this chapter, we establish a stochastic programming based demand estimation method for network level OD demand estimation using multiple sets of observed link flow data. This method simultaneously estimates the expected demand and reconstruct OD trips throughout the observation time intervals. There are several major contributions to highlight. From a conceptual perspective, approaching the problem from a fresh angle based on two-stage stochastic programming (SP) framework, where demand parameter estimation is treated as the first stage decision and demand reconstruction as the scenario-dependent recourse decision, we are able to make connection between the new estimation and the well-studied SP problems. The advantages of conceptualizing the problem from a systematic approach in the SP framework, rather than a problem-specific ad hoc approach in a previous study (Yang et al., 2018), mainly lie in the opportunities this will open up for exploiting the very rich knowledge already established by the stochastic programming community, including solution methods for large-scale problems and modeling choices for incorporating additional risk

preferences. For example, we have demonstrated the computational benefit via exploiting a scenario decomposition methods from the stochastic programming literature that was designed for coping with computational challenges brought by large scenarios (Rockafellar and Wets, 1991; Ruszczyński, 1997; Fan and Liu, 2010). Also, there are various SP modeling choices to incorporate different risk preferences (Ahmed, 2004). In this study, the estimation objective is set to be risk neutral based on expectation, while confidence interval analysis is conducted as a post-model analysis. Building on the mean-risk SP literature, one may consider explicitly incorporating confidence level (corresponding to a critical threshold) as a chance constraint, for which efficient and scalable solution methods are available (Watson et al., 2010). All of these advancements in SP could directly benefit the estimation problem studied here. From a modeling perspective, we emphasize the importance of incorporating transportation network domain knowledge in data-driven estimation processes. As illustrated in the numerical experiments, by incorporating both network topology information and network operation knowledge in the estimation process, the estimation quality can be largely improved. Results from the numerical experiments are promising and demonstrate the effectiveness of the proposed method.

Despite the exciting new estimation model and results reported here, there are several immediate steps for future research. For example, the proposed estimation framework is flexible to incorporate network operational domain knowledge. One may explore other type of traffic assignment rules. Also, we adopted a least squares based loss functions in this study. One may consider applying the proposed framework with other risk functions used in statistics and information theory. Opportunities in sensor technology can also be incorporated in the future to mitigate the uncertainties and noise in observed data. For example, in-motion weight sensors for trains would provide information on link flow directly beyond APC system. Most interestingly, as demonstrated in the numerical experiments, information pieces collected from various components of the network can play different roles for different OD pairs' estimation quality. A more proactive approach to seek critical information needed

to ensure a good estimation quality would be a worthy next step.

# Chapter 3

# Application-Driven Representation Learning for Feature Extraction and Anomaly Detection on Networked Flows

## 3.1 Introduction

Modern transportation systems have benefited from rich spatial and temporal data related to vehicle and pedestrian flows, travel behaviors and demands, and infrastructure performance measurements. For example, a vast amount of traffic and vehicle trajectory/dynamics data is continuously collected by a wide coverage of infrastructure-based and mobile sensors in the highway system and local roadways. However, because of limitation in computing or hardware power and data storage capability, it is nearly impossible to store and process all the data directly in its original high spatiotemporal dimension. Also, repetitive measurements for daily mobility data may be redundant in terms of key information and can lead to waste in storage space and communication bandwidth if all raw data is saved. Therefore, finding

representative underlying patterns for high-dimensional data has attracted much research effort, especially for data with high spatial and temporal correlations, which is common for most mobility-related data.

Representation learning has shown promise in coping with high dimensional data. First, by learning the lower dimensional representation from the original high dimensional data, the learned representation can capture dominant information in the original dataset, thus serving as a denoising procedure. Second, some familiar concepts that work well in low dimensional cases may not transcend to high dimensional situation. For example, the distance measure, which is often needed in data learning algorithms, can be ill-posed in high dimension. As Aggarwal et al. (2001) showed, the $L_k$ norm degenerates fast with increasing dimensionality for higher values of $k$. Domingos (2012) argued that most of the mass of a high-dimensional space is in the corner, not the central area. Thus, the samples are more likely to appear closer to a face of the hypercube than to their nearest neighbor. Representation learning helps address these challenges by revealing key data features and better capturing relationships within data samples.

Representation learning methods can be categorized into linear methods and nonlinear methods. Linear methods assume that the data actually lie on a low dimensional subspace. One popular example is Principal Component Analysis (PCA), where the objective is to maximally preserve data variance in the embedding space (Wold et al., 1987). Low-dimensional representations can be obtained by projecting the high-dimensional data onto the subspaces spanned by a subset of an eigenbasis. Nonlinear methods usually assume the data live in one or a few manifolds. The core assumption driving those methods is the manifold hypothesis, as most relevant information is concentrated in small number of low dimensional manifolds for high dimensional structures. The major difference among nonlinear dimensionality reduction techniques lies in the preservation of various metrics about the distance of the data in the embedding space. For example, Isomap preserves the pairwise geodesic distance along the submanifold (Tenenbaum et al., 2000). Local linear embedding (LLE)

assumes data can be represented as linear combinations of nearest neighbors on a locally linear manifold (Roweis and Saul, 2000). Laplacian eigenmaps uses a graph as the discrete approximation of the low-dimensional manifold (Belkin and Niyogi, 2002). Conformal eigenmaps computes optimal angle-preserving map using partial basis from graph Laplacian (Sha and Saul, 2005). Several recent studies explored deep learning techniques for representation learning, including auto-encoder (Hinton and Salakhutdinov, 2006), convolutional neural network (Shamsolmoali et al., 2019) and others. However, it is still an ongoing research question about how the dominant features are selected in the model training step and how they contribute to the task of interest for the deep learning approaches.

In terms of applications of the general representation learning methods specifically for transportation domain problems, several limitations are observed. 1) **Only structural penalizations are considered.** As widely shown in statistical estimation tasks, the only way to overcome the curse of dimensionality is to incorporate addition assumptions about the regression function beyond the samples themselves (Györfi et al., 2002). However, often times only assumptions on structural penalties are considered for the data analysis, for example $\mathcal{L}1$ penalty in LASSO (Tibshirani, 1996), sparsity penalty on wavelet coefficients (Donoho and Johnstone, 1995) and 'roughness' penalty for smoothing splines (Unser et al., 1993). Limited attention has been paid to how to design a representation learning method taking advantage of domain knowledge. In fact, domain knowledge could provide information about the data generation mechanisms and dominant underlying patterns from knowledge in traffic flow dynamics, network operations, built-environment and so on. 2) **How the data is represented is often treated separately with how the data should be used for downstream applications.** Representation learning techniques have been discussed in the transportation community, especially related to data mining ((Djukic et al., 2012),(Yang et al., 2017) and measurements for vehicles and systems (Saffari et al., 2020). However, the way the compressed data are used in downstream tasks is rarely jointly considered when trying to find a good underlying pattern of the data. The information loss of data itself is usually the

only criterion when designing dimension reduction or data compression algorithms. On the other hand, it is widely recognized that due to complex network behaviors and interactions, different data elements may impact the downstream application performance differently. As a result, generic data-driven approaches that focus only on the data itself might not always provide the most effective data representation and compression schemes for the downstream applications.

In this study, we aim to establish an application-driven representation learning (ADRL) approach to find the dominant features in observed traffic data by incorporating downstream applications into the estimation process. This research design philosophy aligns with the general concept of end-to-end learning, which promotes the integration of various data processing and learning steps/tasks in a holistic manner (Cai et al., 2016; Zhou and Tuzel, 2018). Our proposed conceptual method leads to a constrained optimization problem which could be effectively solved using Stiefel manifold optimization techniques if sparsity exists in the manifold space. Through the implementation of three case studies, besides validating our approach by comparing the quality of results with existing methods, we are particularly interested in exploring the following three questions/hypotheses. First, it is well known that traffic data have strong spatial and temporal features in the original Euclidean space. Do they also have dominant features that live in the manifold to enable effective utilization of manifold optimization techniques? Second, is it true that downstream applications matter for the representation learning of traffic data? Lastly, if the answer to the second question is mixed, what kind of applications and data would matter more to incorporate downstream applications?

The rest of the chapter is organized as follows. In Section 2, we present the ADRL framework, which can be solved by Stiefel manifold optimization approach. Then we use network travel time and vehicular emission estimation to demonstrate the effectiveness for ADRL on data compression in Section 3, and a network flow case study to demonstrate the application of ADRL in anomaly detection in Section 4. The last section concludes this

study with high-level insights and possible future directions.

## 3.2 An Application-Driven Representation Learning Approach

In an abstract manner, the representation learning problem may be stated as follows. Consider a data matrix $X \in \mathbb{R}^{n\times d}$ that includes $n$ samples of $d$-dimensional data, with each row $x_i^\top$ as an observation, one aims to learn an orthogonal transformation matrix $\Phi \in \mathbb{R}^{d\times k}$ (with $k < d$) from $d$-dimensional space to a $k$-dimensional space, so that the low-dimensional representation of the data $Y \in \mathbb{R}^{n\times k}$ can be obtained as $Y = X\Phi$.

$$X = \begin{pmatrix} | & x_1^\top & | \\ | & x_2^\top & | \\ | & \vdots & | \\ | & x_n^\top & | \end{pmatrix} \in \mathbb{R}^{n\times d} \qquad \Phi = \begin{pmatrix} | & | & \cdots & | \\ \phi_1 & \phi_2 & \cdots & \phi_p \\ | & | & \cdots & | \end{pmatrix} \in \mathbb{R}^{d\times k}$$

### 3.2.1 An Application-Driven Representation Learning (ADRL) Model

Consider the following optimization problem,

$$\begin{aligned} \min_{\Phi\in\mathbb{R}^{d\times k}} \quad & \mathcal{Q}\Big(X, Dec(Enc(X))\Big) + \gamma\mathcal{R}\Big(X, Dec(Enc(X))|\mathcal{A}\Big) \\ \text{s.t.} \quad & \Phi^T\Phi = I_{k\times k} \end{aligned} \tag{3.2.1}$$

where $X \in \mathbb{R}^{n\times d}$ is the data matrix. Encoding transformation $\Phi \in \mathbb{R}^{d\times k}$ represents the mapping from original feature space $\mathbb{R}^d$ to low dimensional space $\mathbb{R}^k$. The data reconstruction error is characterized using an encoder-decoder fashion, as $\hat{X} \in \mathbb{R}^{n\times d} = Dec(Enc(X))$. The encoder $Enc(\cdot)$ carries the learned representation of data , the decoder $Dec(\cdot)$ reconstructs the data from the encoding representation. Let $\mathcal{Q}(X, \hat{X})$ represent the information loss in the data itself (from the original data to the reconstructed data). The second term

$\mathcal{R}(X, \hat{X}|\mathcal{A})$ accounts for the impact of the information loss of data on the downstream application. Application metric $\mathcal{A}$ consists of the coefficients determined by a specific downstream application of interest. For now, one may think that $\mathcal{A}$ is known exogenously. Later, we will discuss how to use approximation techniques combined with augmented data to relax this assumption. The weighting parameter $\gamma$ controls the balance of two sources of information loss. The constraint ensures the projection $\Phi$ is orthogonal, to ensure data stability and enable a wide range of other applications, including linear filter for smoothing. An alternative approach is to include the soft shrinkage as in L2 penalty $\|\Phi\|_2^2$ in the objective, which can reduce the solution complexity, but it will sacrifice the orthogonality. The goal is to minimize the error caused by the representation learning process, which includes the error in the data itself as well as the error in the downstream application caused by the information loss of the compressed data.

Following the minimum reconstruction error criteria in the Euclidean sense for the loss functions $\mathcal{Q}$ and $\mathcal{R}$, our proposed framework can be implemented as

$$\begin{aligned} \textbf{ADRL.} \quad & \min_{\Phi \in \mathbb{R}^{d\times k}} \quad \|X - X\Phi\Phi^\top\|_F^2 + \gamma \sum_{i=1}^{n} w_i (x_i^T a_i - x_i^T \Phi\Phi^\top a_i)^2 \\ & \text{s.t.} \quad \Phi^\top \Phi = I_{k\times k} \end{aligned} \tag{3.2.2}$$

The resulting estimates of low-dimensional encoding is obtained as $Enc(X) = Y = X\Phi$. Reconstructed data $\hat{X}$ is cast as the rank-K approximation of the original data, as $\hat{X} = Dec(Enc(X)) = X\Phi\Phi^T$. The minimum data reconstruction error can be written as the squared distortion measure in the Frobenius sense as $\mathcal{Q}(X, Dec(Enc(X))) = \|X - X\Phi\Phi^\top\|_F^2$. For the $i$th sample, the metric for downstream application may be represented using a linear model $J_i = x_i^\top a_i$. Note that we allow the application measure $a_i \in \mathbb{R}^{d\times 1}$ to be sample dependent, which accounts for cases when the downstream performance measure is nonlinear to $x$. Similarly, the application metric for the reconstructed data is $x_i^T \Phi\Phi^\top a_i$. Thus, the error term for the downstream task can be written as $\mathcal{R}(X, = Dec(Enc(X))|\mathcal{A}) = \sum_{i=1}^{n} w_i (x_i^T a_i - x_i^T \Phi\Phi^\top a_i)^2$. Note that other differentiable loss functions can also be applied, such as Pseudo-

Huber loss and Kullback–Leibler divergence. To balance the unequal covariance in the residuals, we associated a weight $w_i$ to each application error term. Motivated by weighted linear regression (Johansen, 1980), we let the weight $w_i$ be a surrogate to the reciprocal of sample covariance estimate

$$w_i = \frac{1}{(x_i^\top a_i - \bar{\mu})^2/(n-1)} \tag{3.2.3}$$

with sample mean estimate of the application measure as $\bar{\mu} = \frac{1}{n}\sum_i^n x_i^\top a_i$. A robust version can also be used as

$$w_i = \frac{1}{|x_i^\top a_i - \bar{\mu}|/n} \tag{3.2.4}$$

### 3.2.2 Numerical Solution

The constrained optimization 3.2.2 is nonconvex and difficult to solve due to the orthogonal constraint $\Phi^\top\Phi = I$. This class of nonconvex problems with orthogonal constraints is of great importance in signal processing and statistical learning communities (Absil et al., 2009; Jiang and Dai, 2015). The feasible set $\mathcal{L}_{d,k} = \{\Phi \in \mathbb{R}^{d\times k} : \Phi^\top\Phi = I\}$ is often referred to as the Stiefel manifold. Similar to many convex optimization methods, an effective manifold optimization procedure usually has two iterative steps: 1) find a tangent vector as the search direction; 2) invoke a retraction that maps the tangent vector onto a point on the manifold. We adopt the manifold optimization method based on Cayley transformation proposed by Wen and Yin (2013). Denote $G = (\frac{\partial \mathcal{F}(\Phi)}{\partial \Phi_{i,j}})$ as the gradient for the differentiable objective function $\mathcal{F}$ with respect to the decision variables $\Phi$ in 3.2.2. The gradient of the objective function can be evaluated in the closed form for two terms as

$$G = G_1 + \gamma G_2 \tag{3.2.5}$$

with the first term

$$G_1 = -2X^\top X\Phi \tag{3.2.6}$$

and the second term

$$G_2 = \sum_{i=1}^{n} w_i[-2(x_i x_i^\top a_i a_i^\top + a_i a_i^\top x_i x_i^\top)\Phi + 2(a_i a_i^\top \Phi\Phi^\top x_i x_i^\top + x_i x_i^\top \Phi\Phi^\top a_i a_i^\top)\Phi] \tag{3.2.7}$$

Define the gradient of the objective function in the tangent space as

$$\nabla \mathcal{F} = S\Phi \tag{3.2.8}$$

where $S = G\Phi^\top - \Phi G^\top$ is a skew symmetric matrix as one metric of the tangent space of the manifold $\mathcal{L}_{d,k}$. The iterative update scheme can be determined by the Crank-Nicholson-like scheme, which smoothly maps a tangent vector to the manifold, as

$$\mathcal{Y}(\tau) = \Phi - \frac{\tau}{2} S(\Phi + \mathcal{Y}(\tau)) \tag{3.2.9}$$

where $\mathcal{Y}(\tau)$ is the curve on the manifold, $\tau$ is the step size in the descent updating path. The updating scheme $\mathcal{Y}(\tau)$ can be evaluated in the closed form, known as Cayley transformation

$$\mathcal{Y}(\tau) = (I + \frac{\tau}{2} S)^{-1}(I - \frac{\tau}{2} S)\Phi \tag{3.2.10}$$

The Cayley transformation possesses nice properties in that 1) the updating curve $\mathcal{Y}(\tau)$ is smooth in $\tau$; 2) the new trial point stays in the feasible set $(\mathcal{Y}(\tau))^\top \mathcal{Y}(\tau) = \Phi^\top \Phi$ for all $\tau \in \mathbb{R}$; and 3) $\frac{d}{d\tau}\mathcal{Y}(0)$ equals the projection of the negative gradient direction $-G$ into the tangent space of the manifold $\mathcal{L}_{d,k}$ at $\Phi$. Next, the iterative procedure can be implemented with these core steps:

**Algorithm 3:** Curvilinear Search based Gradient Descent Method

**Result:** $\Phi^*$
Initialization: $k \leftarrow 0$, ;
**while do**
  Evaluate gradient $D \leftarrow \nabla F(\Phi_k)$;
  Compute $S \leftarrow D\Phi^\top - \Phi D^\top$;
  Choose step size $\tau_k$ call curvilinear line search module;
  Compute update rule $Y(\tau_k) = B\Phi \quad with \quad B = (I + \frac{\tau_k}{2} S)^{-1}(I - \frac{\tau_k}{2} S)$;
  Update $\Phi_{k+1} = Y(\tau_k)$
  Stopping criteria: **if** $\|\nabla F(\Phi_k)\| \geq \epsilon$, *where* $\nabla F(\Phi_k) = S\Phi$ **then**
    STOP
  **else**
    $k \leftarrow k + 1$
  **end**
**end**

**Algorithm 4:** BB Curvilinear Search

**Result:** $\tau^*$

Initialization: $k \leftarrow 0$, $\rho_1, \delta, \eta, \epsilon \in (0, 1)$;

**while** $\|\nabla \mathcal{F}(\Phi_k)\| \geq \epsilon$ **do**

`// Curvilinear search`

**while** $\mathcal{F}\left(Y_k(\tau)\right) \geq C_k + \rho_1 \tau \mathcal{F}'\left(Y_k(0)\right)$ **do**

$\tau \leftarrow \delta\tau$

**end**

`// Update scheme`

$\Phi_{k+1} \leftarrow Y_k(\tau)$;

`// Regulating step size`

$Q_{k+1} = \eta Q_k + 1$ ;

$C_{k+1} \leftarrow \left(\eta Q_k C_k + \mathcal{F}\left(X_{k+1}\right)\right)/Q_{k+1}$ ;

Set $\tau \leftarrow \max\left(\min\left(\tau_{k+1}, \tau_{max}\right), \tau_{min}\right)$: Choose closet point to $\tau_k = \frac{\mathrm{tr}\left((T_{k-1})^\top T_{k-1}\right)}{\left|\mathrm{tr}\left((T_{n-1})^\top U_{k-1}\right)\right|}$ in the interval $[\tau_{min}, \tau_{max}]$, with $T_{k-1} = \Phi_k - \Phi_{k-1}$ and $U_{k-1} = \nabla\mathcal{F}(\Phi_k) - \nabla\mathcal{F}(\Phi_{k-1})$

**end**

## 3.3 Feature Extraction

### 3.3.1 Case Study 1: ADRL - Network Travel Time

In this section, we present the first case study with ADRL on network travel time. Consider the data as the OD flow $X \in \mathbb{R}^{n\times d}$, where $x_i \in \mathbb{R}^d$ represents one observation of OD flow for all given OD pairs in the network. Link flow $v_i$ can be computed as a result of traffic assignment $H_i$. In this case study, we adopt the user equilibrium (UE) traffic assignment as an illustrative example to compute $H_i$ exogenously. In UE, for each OD pair, all used routes have equal and minimal travel time. Other traffic assignments can also be applied here, including system optimal (SO), stochastic user equilibrium (SUE), and all-or-nothing assignment (AON). For link cost function in the UE program, we take the classic BRP link performance function. The link travel time $t_e$ for link $e$ in the network can be computed as

$$t_e = f_e(1 + \alpha(\frac{v_e}{\kappa_e})^\beta) \tag{3.3.1}$$

where $f_e$ is free flow travel time, $v_e$ is link flow, $\kappa_e$ is link capacity, $\alpha, \beta$ are coefficients. As a result of UE, we have OD-link network assignment as a linear system

$$v_i = H_i x_i \tag{3.3.2}$$

where $H_i \in \mathbb{R}^{l \times d}$ is linear proportional mapping of traffic assignment from OD flow $x_i \in \mathbb{R}^d$ to link flow $v_i \in \mathbb{R}^l$. Denote $c_i \in \mathbb{R}^l$ as the vector of network travel time, with each element being the travel time $t_e$ computed using the BPR function in Eq.(3.3.1) for each link based on the link flow $v_i$. The application metric as network cost coefficients can be written as $a_i = H_i^\top c_i$. Therefore, the total network cost can be cast into a linear function for the $i$th observation of OD flow $x_i$,

$$x_i^\top a_i = x_i^\top H_i^\top c_i \tag{3.3.3}$$

The same cost coefficients $a_i$ is applied onto the reconstructed OD flow $x_i^\top \Phi\Phi^\top$ for corresponding elements. Note that in this case study, $a_i$ is data dependent because both $H$ and $c$ vary as the OD flows change.

Next, we show an example network modified based on the Sioux-Falls network (Suwansirikul et al., 1987), in Figure 3.1. We allow three high density central business districts (CBDs) centered around nodes 8, 11 and 22. The link performance functions for the CBD regions are adjusted to be more sensitive to congestion to mimic the active stop-and-go and pedestrian crossing movements. The OD flow $X$ is synthesized following groups of multivariate normal distributions with several block structures in covariance matrix. The sample size $n$ is 400, dimension $d$ (number of covariates as OD pairs) is 552 and low dimensional budget $k$ is 30. The data $X$ and sample covariance matrix of $X$ are shown in Figure 3.2. The network costs $a_i$, for $i = 1, \ldots, n$ are stacked into the matrix form as column vectors. The deviations of $a_i$ from their mean vector are shown in Figure 3.3. Note that the deviations are relatively small in scale, because the collective routing behavior is relatively stable even given fluctuations in OD demand in the network. However, what is worth noting is the variations among the OD pairs (rows of A matrix), such that different OD pairs would

have different contributions to the total network travel time, which is the second term of the objective function in the optimization program 3.2.2.

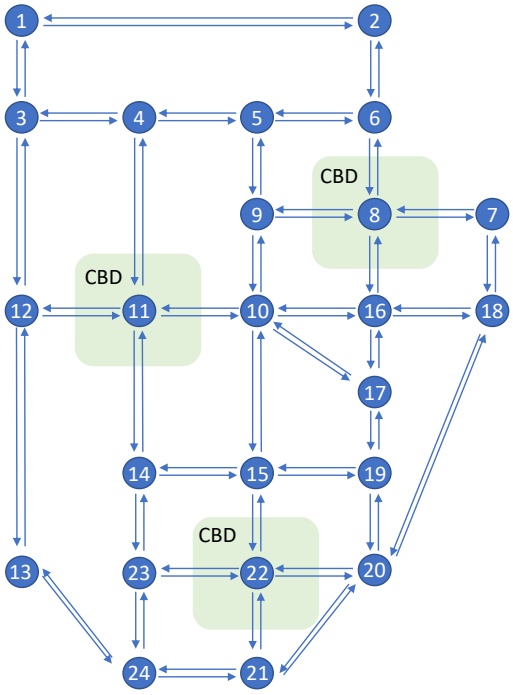



Figure 3.1: Modified Sioux-Falls network

First, we compare the transformation matrix obtained from ADDR $\Phi$ with the PCA transformation $P_k$, shown in Figure 3.4. There are major differences for the first dominant components (columns) in subplot (a). This is due to the fact that PCA implicitly associates higher weights on the first computed components. For example, the power method iteratively computes the dominant eigenvectors according to the order of deceasing eigenvalues $|\lambda_i|$. Differently, our ADRL approach allows same weights for all dominant components, i.e. they are simultaneously computed via the manifold optimization. Therefore, we have decreasing differences over the columns of the transformation matrices $\Phi - P_k$. The reconstruction errors (shown in subplot (b)) of the first term in the objective function for program 3.2.2 computed from ADRL and PCA show a similar behavior. However, for the second term regarding the

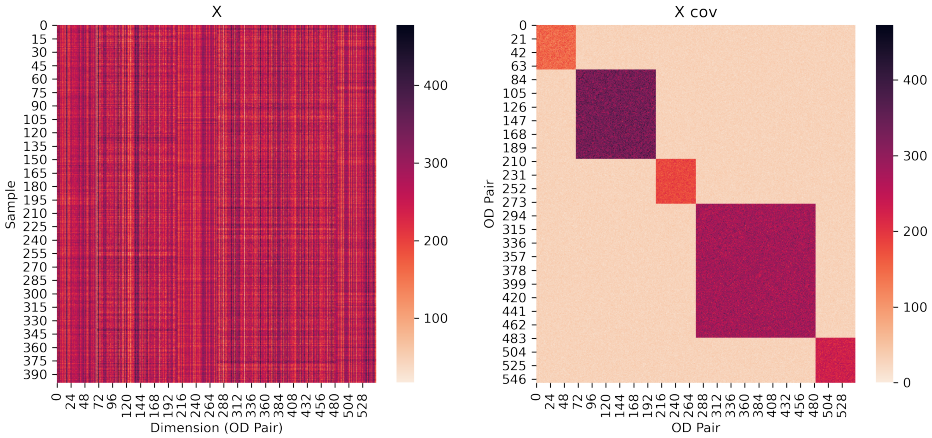



Figure 3.2: OD flow and its sample covariance

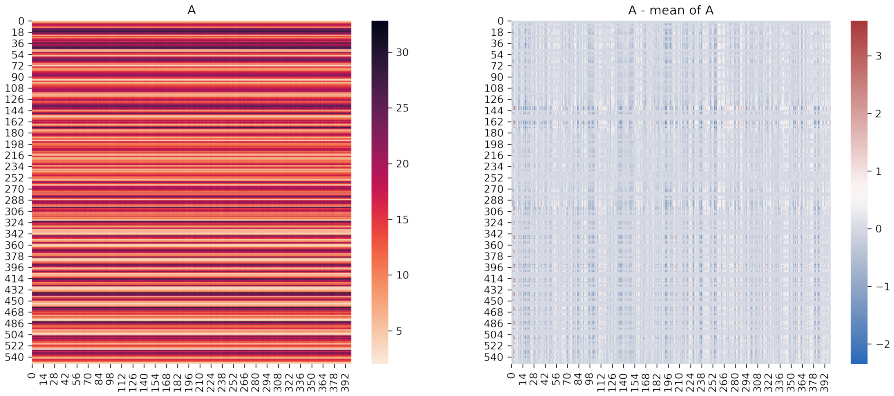


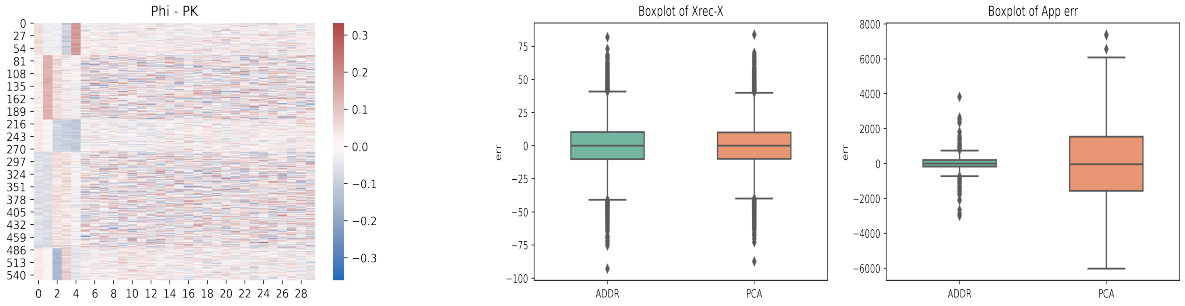

Figure 3.3: Application metric and deviations from mean values

application distortion, PCA yields a much higher error than ADRL. This result indicates the benefit of incorporating application-driven performance measures in the design of learning approaches as emphasized in this study.

Next, we show the convergence results for ADDR using the Stiefel manifold optimization technique. Figure 3.5 plots the convergence of the solution algorithm in terms of 1) objective function, 2) $\nabla\mathcal{F}$ and 3) changes in $\Phi$. The zoomed-in first 50 iterations are also attached for each subplot. The speed of convergence is fast for the first several iterations, and then

slows down afterwards. Based on multiple runs on a Macbook Pro machine with 8 core CPU and 16 GB memory, the first 50 iterations took around 2.5 minutes on average, where the majority of the optimal results were reached. The average computation time until the final convergence (around iteration 1,700) was 73 minutes.

Finally, we present the sensitivity results against $\gamma$ in Figure 3.6 and dimension budget $k$ in Figure 3.7. When the weighting parameter $\gamma$ increases, the model emphasizes more the application based cost in the objective 3.2.2. Therefore, we observe deceasing cost associated with the second term and increasing cost with the first term. Due to the magnitude of these two terms, the optimal objective values is still driven mainly by the first term. When the low dimension budget $k$ increases, we observe decreasing optimal objective loss function, since more information can be preserved.




(a) ADDR Phi vs PCA

(b) Reconstruction error

Figure 3.4: ADRL vs. PCA

### 3.3.2 Case Study 2: ADRL - Emission

In the second case study, we present the application ADRL in emission estimation. In general, emission models can be categorized into macroscopic models and microscopic models. Macroscopic emission models generally use vehicle average speed and vehicle miles traveled (VMT) for the large network emissions estimation, for example the Emission Factors Model

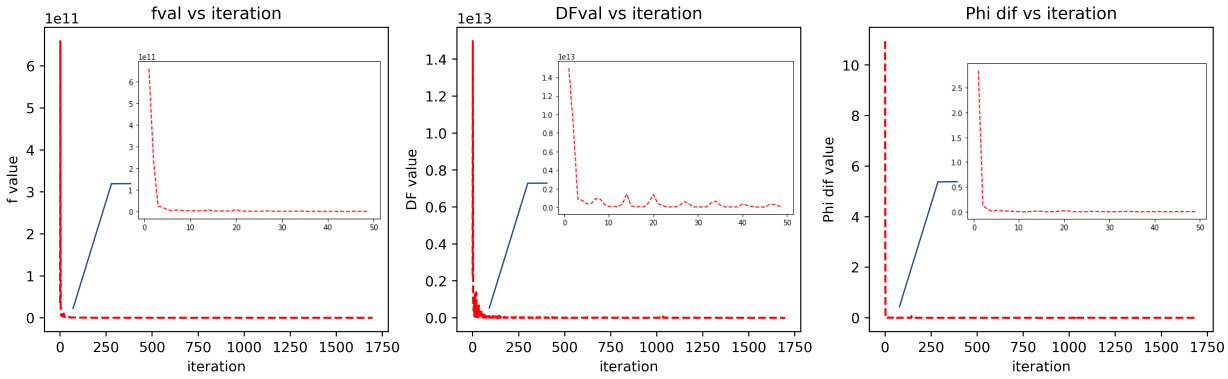



Figure 3.5: Convergence plot

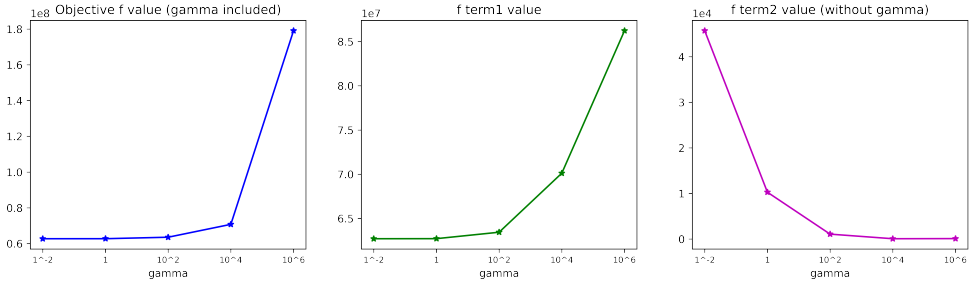



Figure 3.6: Optimal objective values vs. weighting parameter $\gamma$

(EMFAC) developed by California Air Resource Board (CARB).[1] Microscopic models are aimed at disaggregated measures, including vehicle dynamics and instantaneous speed, for example, Motor Vehicle Emission Simulator (MOVES)[2] and Comprehensive Modal Emissions Model (CMEM) (Barth et al., 2000).

Typically, a macroscopic emission model would be input with aggregated speed distribution (or VMT by speed bins and spatial regions) to save computational burden at a price of sacrificed accuracy. In this case study, we demonstrate that ADRL approach can provide the compressed low dimensional data on vehicle instantaneous speeds to enable accurate quantification of vehicular emission using EMFAC model. The observed vehicle trajectory

[1]https://arb.ca.gov/emfac/
[2]https://www.epa.gov/moves

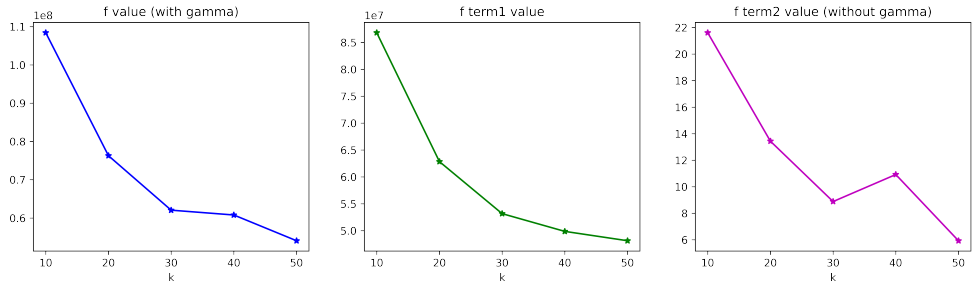



Figure 3.7: Optimal objective values vs. dimension budget $k$

data are obtained from NGSIM [3]. Figure 3.8shows the relationship between common criteria pollutants and vehicle speeds in EMFAC model. The emission factors differ between regions as they are estimated based on characteristics by region. Here we use factors from Alameda County for illustration purpose. Figure 3.9 shows the trajectory data of a segment on I-80 retrieved from NGSIM (only the auto mode from NGSIM dataset is included for this analysis).

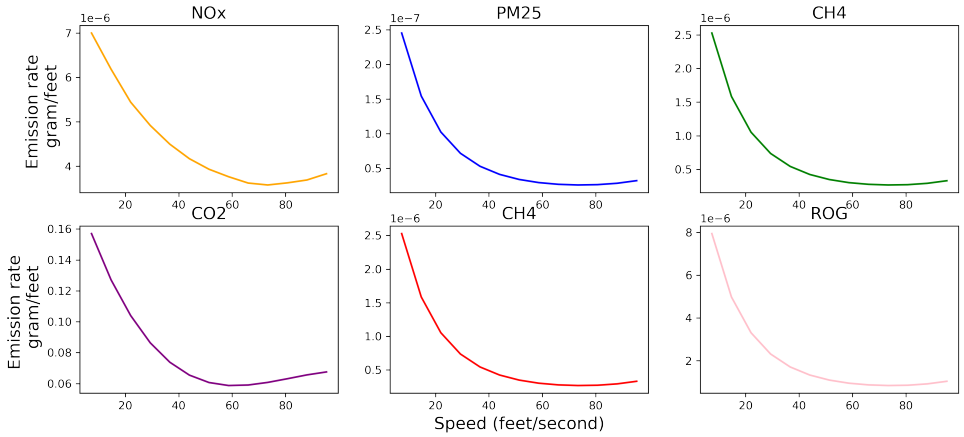



Figure 3.8: Emission rates

Next, we implement the ADRL method to the vehicular emission estimation application. We discretize the road section into spatial intervals of feet length. The speed data are

[3]https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm

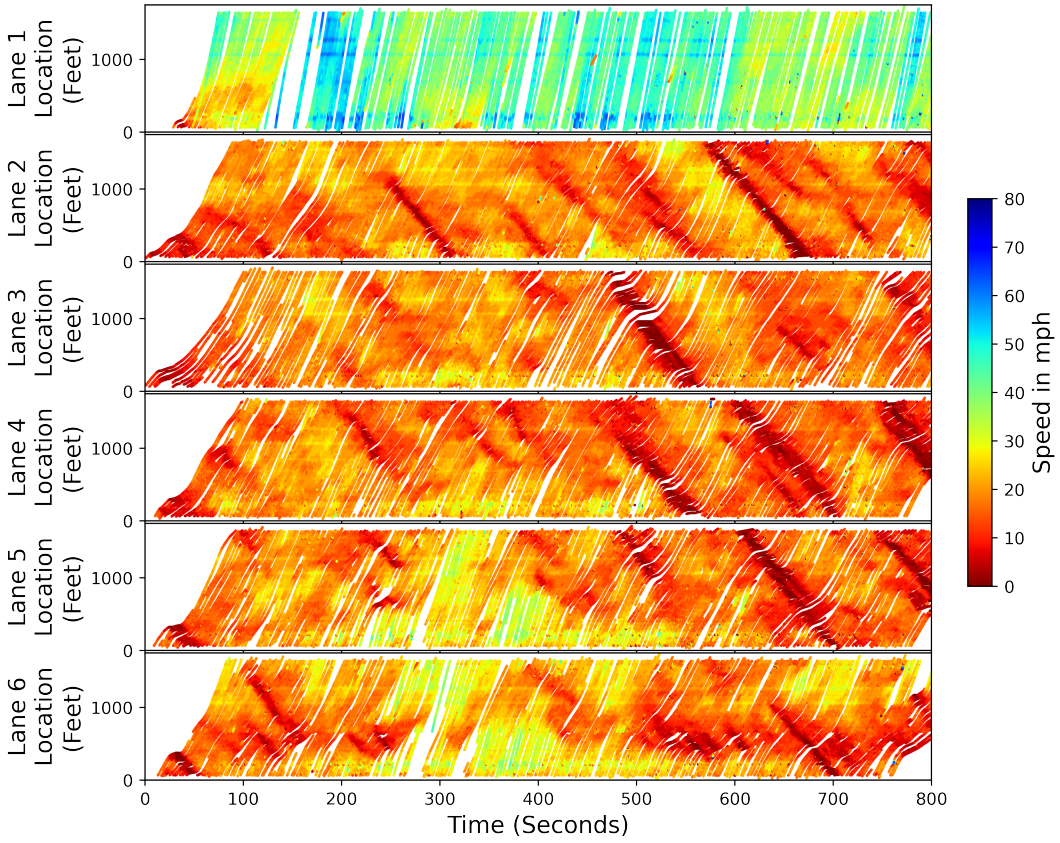



Figure 3.9: Vehicular trajectories with instantaneous speeds (inspired by Li et al, 2020)

mapped to these internals, where missing values or multiple values are imputed using linear interpolation of the nearest observations. Thus the resulting speed data matrix $X$ represents the instantaneous speed (mph) of all $n$ vehicle samples by $d$ spatial intervals (feet) in the roadway section. The emission metric $A$ represents emission rates (g/mile) by instantaneous speed (mph).

Emissions are generally not a linear function of speed. Take PM2.5 as an example, the nonlinear relationship between emission rates and speed can be represented as a polynomial function,

$$y = -1.350 \times 10^{-8}x + 2.000 \times 10^{-10}x^2 - 0.955 \times 10^{-13}x^3 + 3.228 \times 10^{-7} \tag{3.3.4}$$

where $x$ is the speed (feet/second) and $y$ is the emission rate of PM2.5 (milligram/feet).

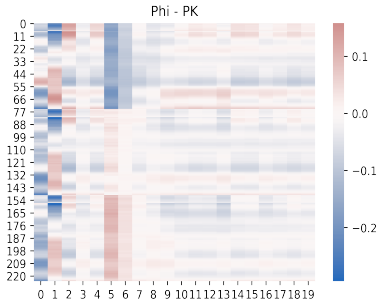
Our proposed method is flexible to accommodate nonlinear performance measure through construction of an augmented data matrix

$$\tilde{x}_i = [x_i^J, x_i^{J-1}, \ldots, x_i] \tag{3.3.5}$$

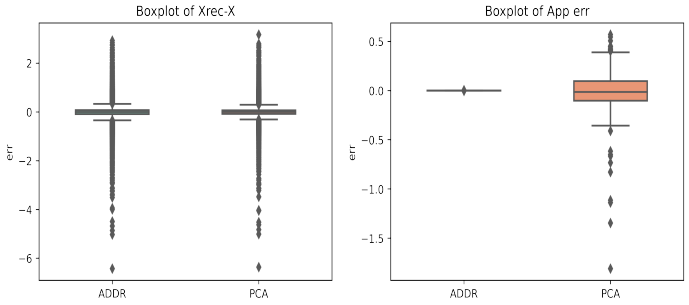
The nonlinear function for quantifying emissions can be written as

$$z_i = \tilde{x}_i \beta \tag{3.3.6}$$

With this setting, for PM2.5, $a_i = [\beta_1, \beta_2, \beta_3]^\top$. Note that if the downstream performance measure is in other nonlinear form, one may use a polynomial regression model to approximate it.




(a) ADRL Phi vs PCA




(b) Reconstruction error

Figure 3.10: EM ADRL vs. PCA

We standardize the input data $X$ for ADRL to mitigate the unequal variance across the covariates. The results obtained for standardized data can be reconstructed by adding

back the mean value and multiplying the covariance. The comparison of ADDR and PCA is shown in Figure 3.10. The difference in the estimated emission from the two models is negligible (at about $4.6 \times 10^{-10}$ milligram). The estimated total emission 1790563.008 milligram from both models is identical to the total emission computed from the original data. The similarity between results from ADRL and PCA is not surprising for this application. Unlike the previous case study where network interactions take place to cause variations in elements of matrix $A$ across different samples and different OD pairs, the relative contribution to the total reconstruction error, $a_i$, does not change between samples or road sections. Therefore, preserving information for the downstream application metric would be close to the information in the data itself. This result reinforces our emphasis on application-driven representation learning in problems with network structure, where variables interact with each other in a more complex manner and are often not easily separable.

## 3.4 ADRL based Anomaly Detection

A primary goal in anomaly detection is to separate data variation and noise within normal instances from true anomaly instances (Pimentel et al., 2014). Thus, it is typically not recommended to apply anomaly detection directly on the original dataset in the high dimensional case as data variation and observation noise would negatively impact the effectiveness. Therefore, anomaly detection can be performed onto the low dimensional data since they capture the dominant information in the data and act as a denoising procedure. For more comprehensive reviews of the anomaly detection problem, see, for example, Chandola et al. (2009) and Agrawal and Agrawal (2015). There are four main categories of anomaly detection methods:

- Probabilistic density based methods: build on generative probability density function of the data (e.g. Extreme Value Theory (Clifton et al., 2011) and kernel density estimation (Subramaniam et al., 2006);
- Distance based methods: build on distance or similarity to local dense neighborhood

or cluster (e.g. k-nearest neighbor KNN graph (Hautamaki et al., 2004));

- Subspace and reconstruction based methods: build on the underlying representation of the data (e.g. kernel PCA (Hoffmann, 2007) and self-organising network (Marsland et al., 2002)); and
- Information theoretic methods: build on the information content of the data (e.g. Kolmogorov complexity (Keogh et al., 2004)).

In this section, we demonstrate the capability of our ADRL method for anomaly detection using the problem context described in Case Study 1 as an example. Unlike existing anomaly detection studies that build on data features obtained from established dimension reduction methods, the low dimensional representation from ADRL captures relationship among OD flows $X$ and their impacts on the overall network performance measured by the total travel time. Later, the experiments conducted in this section suggest that anomaly detection based on data X alone are more likely to lead to 'false alarm' of false positive anomaly instances. Adding the application metric from domain knowledge, i.e. using the learned representation from ADRL, could serve as a regularizer to amplify the difference between the normal and anomaly instances.

Different types of supervision exist for anomaly detection: 1) **Supervised**: Training data consists of 'normal' and 'anomaly' instances, and are labeled with 'normal' or 'anomaly' accordingly. The labels of test data are unknown and need to be determined. 2) **Unsupervised**: Training data consists of 'normal' and 'anomaly' instances, but no label is given in prior for either training set or testing set. 3) **Semi-supervised** : Training data consists of only 'normal' instances or only anomaly instances, and test data contains both 'normal' and 'abnormal' instances. This is suitable for a situation where previously unobserved patterns (not included in training data) need to be learned and labeled for new observations in the test dataset. We adopt the semi-supervised setting here, with an assumption that the training data are free of anomaly samples and testing data include a combination of 'normal' and 'abnormal' data points. The semi-supervised setting is also referred as novelty detection in

scikit learn [4].

Given a data point $y_i$, we define the anomaly score as the average distance between $y_i$ and its $k$ nearest neighbors

$$S(y_i) = \frac{1}{k}\sum_{j=1}^{k} d(y_i, y_i^j) = \frac{1}{k}\sum_{j=1}^{k} \|y_i - y_i^j\| \tag{3.4.1}$$

Here we let $d(\cdot)$ be the Euclidean distance, other measures can also be used, including Manhattan distance and Chebyshev distance. Our method depends on the following two widely accepted assumptions in the literature. **Assumption 1**: Data can be embedded into lower dimensional subspace where normal instances and abnormal instances are significantly different. **Assumption 2**: Normal data instances occur in dense local neighborhoods of the dataset, and anomaly instances are far from these dense regions. Our ADRL based KNN anomaly detection is described in Algorithm 5.

**Algorithm 5:** ADRL based KNN anomaly detection

**Input:** Training data $X_{tr} \in \mathbb{R}^{n\times d}$, testing data $X_{te} \in \mathbb{R}^{m\times d}$, application metric $A$ ;
**Result:** Transformation $\Phi_{tr} \in \mathbb{R}^{p\times p}$, low-dimensional data $Y_{tr} \in \mathbb{R}^{n\times p}$, $Y_{te} \in \mathbb{R}^{m\times p}$ anomaly label $z_i = \{0, 1\}, \forall 0 \leq i \leq m$
Compute $\Phi_{tr}$ from ADDR equation 3.2.2 using $X_{tr}$ and $A$;
Obtain low-dimensional data as $Y_{tr} = X_{tr}\Phi_{tr}$, $Y_{te} = X_{te}\Phi_{tr}$;
Compute KNN distance matrix $D_y\{y_{te}, y_{tr}\} \in \mathbb{R}^{(n+m)\times(n+m)}$ using the distance metric $d_y(y_i, y_i^k)$ for all $y_i$ as anomaly score; Distance (anomaly score) criteria: $y_i = 1$ if $D_y(i, j) > \tilde{d}$; $y_i = 0$ otherwise;

Let us now compare our ADRL-KNN based anomaly detection approach with existing KNN-based approaches, including the KNN base approach (which uses the KNN distance metric for data $X$ in the original space) and the PCA-KNN approach (which uses the KNN distance metric after PCA transform of the data). Note that for the semi-supervised experiments, we omit the confusion matrix often used in binary classification setting and instead present the distribution of the anomaly scores to better highlight the separability of the normal and anomaly samples.

[4] $https://scikit-learn.org/stable/modules/outlier_detection.html$

First, we show that ADRL can correctly identify the anomaly samples, as true positives. Anomaly cases with extremely high demand in the city's CBD areas were created and added to the normal samples. The data is shown in 3.11, in terms of its original form $X$, lower-dimensional representation $Y$ from ADRL, and $Y$ from PCA. Figure 3.12 and Figure 3.13 show the density of anomaly scores computed using the original data $X$ and the lower-dimensional representation $Y$, respectively. It is reassuring to see that our result is consistent with PCA-KNN result in the context of binary classification of anomaly instances, with a slightly better separability compared to PCA-KNN. Even though we aim for an approach that will provide better estimation for the downstream application, it would be concerning if the binary classification result is too sensitive against one's choice of the dimension reduction approach.

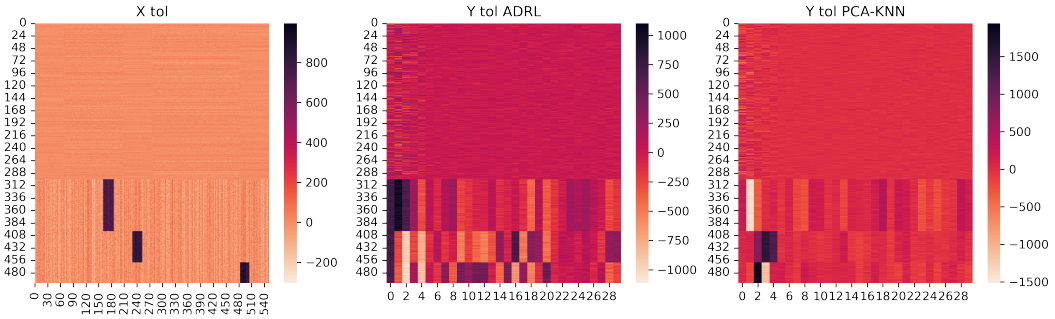



Figure 3.11: Data for X, Y using ADRL and Y using PCA

In addition, we compare the ADRL based anomaly detection with other open-source tools available in PyOD library (Zhao et al., 2019). As reported in Table 3.1, our results are consistent with the ground truth and mostly consistent with other methods. Note that different methods use different anomaly criteria and have different ways to compute the score functions, so we only report the predicted labels for these methods.

Second, we show that ADRL can reduce the noise in true negatives. We add small perturbations $\mathcal{N}(0, 5^2)$ to normal instances, and then check whether these false anomaly instances might be misclassified as false positive. The data is shown in Figure 3.14, with

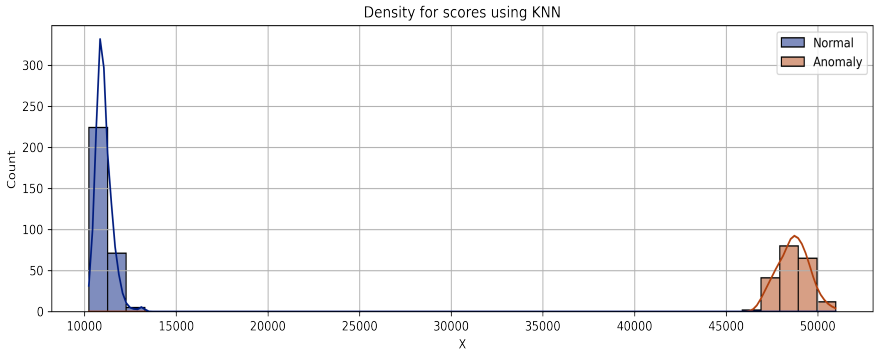



Figure 3.12: Density of Anomaly Scores Using X

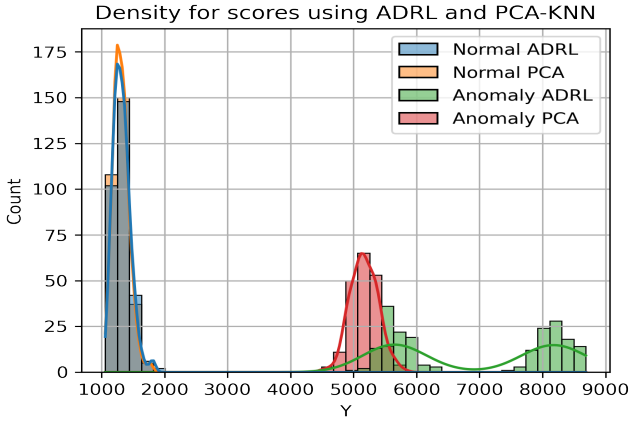



Figure 3.13: Density of Anomaly Scores Using Y

the first 400 samples (rows) for normal instances and the bottom 200 samples (rows) for false anomaly instances. The three panels in Figure 3.14 correspond to the original data $X$, the low dimensional representation $Y$ computed using our proposed ADDR, and the low dimensional representation $\tilde{Y}$ computed using PCA-KNN baseline. Figure 3.15 shows the density of anomaly scores computed using original data $X$ and Figure 3.16 shows the density of scores computed using PCA and our ADDR approach. Clearly, there is a larger overlaps for the anomaly scores using low-dimensional data compared to the result using the original data $X$. This is consistent with the common understanding that a carefully designed low dimensional representation could potentially reduce the risk of false positives in the binary

| Method | # of Inliers | # of Outliers |
|---|---|---|
| **True Label** | 300 | 200 |
| **ADRL** | 300 | 200 |
| Angle-based Outlier Detector (ABOD) | 295 | 205 |
| Cluster-based Local Outlier Factor (CBLOF) | 300 | 200 |
| Histogram-base Outlier Detection (HBOS) | 300 | 200 |
| Isolation Forest | 300 | 200 |
| Feature Bagging | 306 | 194 |

Table 3.1: Comparison with anomaly detection methods

classification of anomaly incidents.

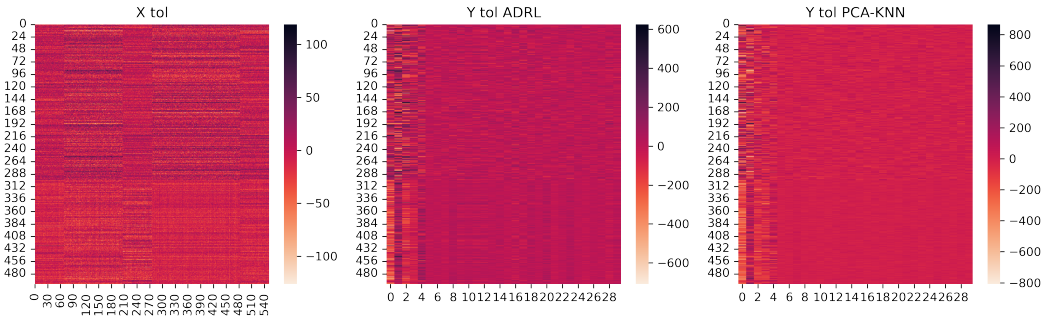



Figure 3.14: Data for X, Y from ADDR and Y from PCA

## 3.5 Discussion

In this chapter, we have established a new representation learning framework that incorporates consideration of data usage in the downstream application in the methodology design. This study has expanded the literature on data-driven methods in transportation science to support the widely recognized need for end-to-end data analytics. Besides representation learning focused in this study, the philosophy aligning with end-to-end analytics may be transferable to data clustering, classification and other similar tasks. The promising results from the numerical examples demonstrated that the traffic data we tested possess sparsity and dominant features in the manifold, for which manifold optimization techniques can be

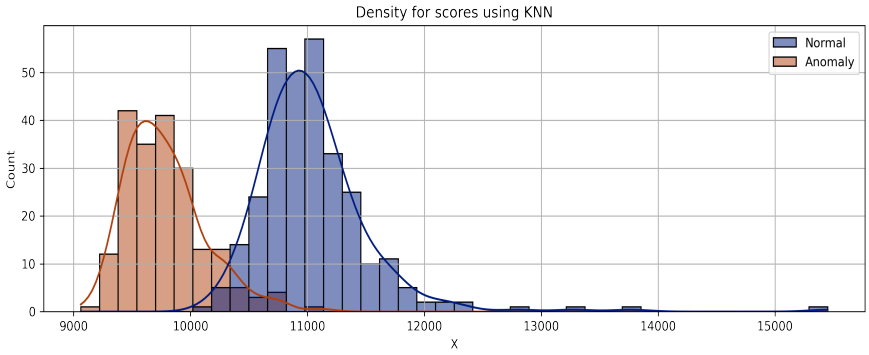



Figure 3.15: Density of Anomaly Scores from KNN (Using X)

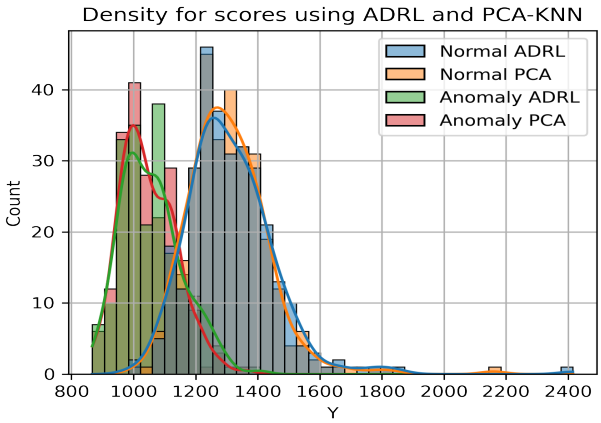



Figure 3.16: Density of Anomaly Scores from PCA and ADRL Using Y

effective. We also learned that adding application-driven terms in the loss function would matter more significantly when the application metric varies with the data. This is often the case for most large-scale transportation network problems, where data pieces interact with each other over a complex network structure. On the other hand, the same data might be used for several different applications. An interesting question would be how to draw some general rules or insights for feature representation for a certain class of problems. Also, how would the application metric or network properties impact the representation selection? What are the critical features that should be preserved despite applications? These will be worthy questions for future efforts.

# Chapter 4

# Continuous-time Markov Chain based Filtering on Directed Graphs

## 4.1 Introduction

The growing interest in statistical machine learning on irregular domain has attracted much attention in many fields, including signal processing, transportation and geospatial systems, and social and behavior analysis. The use of graph structures for data analysis provides a natural way to encode the relationships or similarities between data points, offering many benefits for tasks such as noise reduction and feature extraction. Standard statistical smoothing techniques for noise reduction operate through estimating an underlying signal from noisy observations, often by moving averages or spline regression (Györfi et al., 2002). However, one of the limitations lies in the fact that the localized correlation and heterogeneity of the data samples are often not well captured.

Recent advances in graph machine learning and graph signal processing have demonstrated the effectiveness of utilizing the underlying graph structure to reveal patterns in the data, as discussed in several review papers on graph signal processing (GSP) (Ortega et al., 2018), graph neural networks (Zhou et al., 2020) and network embedding (Cui et al.,

2018). Key to the development of these techniques has been the increasing recognition of the importance of the underlying graph structure in understanding data patterns. Many of these methods are based on the adjacency matrix $A$ and graph Laplacian $L$, rooted from foundational concepts from algebraic graph theory (Chung, 1997; Godsil and Royle, 2001). The adjacency matrix $A$ provides a binary representation of the relationships between the nodes in the graph. The Laplacian graph, a matrix representation of a graph encoding the degree of each node and the connectivity between nodes, can be used to capture the local relationship between data measured on the nodes.

When natural graphs are available at hand, such as sensor networks and highway networks, one can directly obtain the graph structures. Furthermore, the construction of proximity-type or KNN-type (K-Nearest Neighbor) graphs enables the broader application of these methods to more general point cloud data (Maier et al., 2008; Jebara et al., 2009). This opens up new avenues for nonparametric regression and estimation on graphs, with the goal of choosing optimal functions over the graph from observed data.

Several techniques have been proposed for linear smoothing on graphs, including spectral filters (Defferrard et al., 2016) and spatial filters (Subbian and Banerjee, 2013). Spectral graph filters operate in the spectral domain, leveraging the eigenvalues and eigenvectors of the graph Laplacian or other graph matrices to filter signals. On the other hand, spatial graph filters operate directly in the graph domain and filter signals based on their spatial neighborhood relationships. Each of these methods offers unique advantages, and the choice between them often depends on the specific characteristics of the data and the task at hand.

In this paper, we approach these methods from the signal processing standpoint and summarize them into analysis and synthesis classes, following the discussion on (Chen et al., 2001; Elad et al., 2007). The first class of filtering is the so-called analysis framework. In analysis, the regularization or constraint is applied to the fitted signal across nodes. For undirected graphs, the graph Laplacian and its associated eigenvectors have been widely used in graph signal processing for filtering, regression, and many other applications. In

the context of nonparametric regression, variation-based constraints are often introduced to preserve the intrinsic structure of the data. One seminal example is Laplacian-based smoothing where the graph Laplacian is used to approximate the low-dimensional manifold (Smola and Kondor, 2003). A graph signal $\boldsymbol{f} : \mathcal{V} \to \mathbb{R}^N$ is a real-value function defined on the graph vertex domain $V$. Denote $f_n$ as the signal value at the $n$-th vertex $v_n \in \mathcal{V}$. The Laplacian regularization can be expressed the divergence of neighboring data samples on the graph globally

$$\begin{aligned} S_2(\mathrm{f}) &= \frac{1}{2} \sum_{i \in V} \sum_{j \in \mathcal{N}_i} W_{i,j} [f_j - f_i]^2 \\ &= \sum_{(i,j) \in \mathcal{E}} W_{i,j} [f_j - f_i]^2 \\ &= \mathrm{f}^\top L \mathrm{f}. \end{aligned} \tag{4.1.1}$$

with the undirected combinatorial Laplacian defined as $L = D - W$, for degree matrix $D$ and weighted node-node adjacency matrix $W$. Thus, the corresponding $\ell_2$-based Laplacian smoothing can be written as

$$\min_{x \in \mathbb{R}^n} \|f - x\|_2^2 + \lambda x^\top L x \tag{4.1.2}$$

Other regularization has been used for signal denoising. For example, Chambolle (2004) and Chambolle and Pock (2011) used total variation denoising for image denoising and zooming, as

$$J(u) = \sum_{1 \le i,j \le N} \left| (\nabla u)_{i,j} \right|, \tag{4.1.3}$$

with the discrete gradient operator $(\nabla u)_{i,j} = \left( (\nabla u)^1_{i,j}, (\nabla u)^2_{i,j} \right)$ and

$$\begin{aligned} (\nabla u)^1_{i,j} &= \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < N, \\ 0 & \text{if } i = N, \end{cases} \\ (\nabla u)^2_{i,j} &= \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < N, \\ 0 & \text{if } j = N, \end{cases} \end{aligned} . \tag{4.1.4}$$

In another example, Wang et al. (2015) generalized the $\ell_1$ trend filtering to the undirected graphs, motivated by piecewise polynomial approximation to the graph signal. The intro-

duction of the $\ell_1$ penalty of the difference operator helps to exploit local adaptivity with varying local degree of smoothness.

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2} \|f - \beta\|_2^2 + \lambda \left\| \Delta^{(k+1)} \beta \right\|_1 \tag{4.1.5}$$

where $\Delta^{(k+1)}$ is the order-$k+1$ graph difference operator, constructed based on the recursion of first order oriented incidence matrix of undirected graph $G$, as

$$\left\| \Delta^{(1)} \beta \right\|_1 = \sum_{(i,j) \in E} |\beta_i - \beta_j| \tag{4.1.6}$$

and

$$\Delta^{(k+1)} = \begin{cases} \left(\Delta^{(1)}\right)^\top \Delta^{(k)} = L^{\frac{k+1}{2}} & \text{for odd } k \\ \Delta^{(1)} \Delta^{(k)} = D L^{\frac{k}{2}} & \text{for even } k \end{cases} \tag{4.1.7}$$

The second class of methods approach the filtering from the synthesis framework, where one would first construct a suitable basis over the domain and regress the observed signals over the basis. In the more classical setting when graph structure is not incorporated, methods such as splines can be used for filtering in the Euclidean domain

$$\min_{\beta \in \mathbb{R}^n} \|f - \Phi\beta\|_2^2 + \lambda \beta^\top \Omega \beta \tag{4.1.8}$$

where $\Phi$ is the basis matrix and penalty matrix $\Omega$ measures the roughness of the functions. When a graph structure is incorporated, for example, Sharpnack et al. (2013) constructed the (undirected) graph wavelet basis based on the spanning tree. The filtering is given as

$$\min_{\beta \in \mathbb{R}^n} \frac{1}{2} \|y - W\beta\|_2^2 + \lambda \|\beta\|_1 \tag{4.1.9}$$

The graph wavelet basis $W$ is capable of capturing spatially localized smoothness at different locations in the input domain.

From a signal processing viewpoint, the emerging field of GSP is one example of taking the underlying graph structure into the design of the basis on data (Shuman et al., 2013; Ortega et al., 2018). GSP can be seen as a decomposition based approach, where signals

defined on graphs can be decomposed based on their graph harmonics. While dimensionality reduction methods mainly turn the relationship between data samples into a matrix for further analysis, graph signal processing (GSP) techniques treat the networked data as an input to the graph-based filter. The graph shift operator is central to many GSP methods, as it act as the basis of all shift-invariant linear filtering operations on graphs. Shuman et al. (2013) defined the graph Fourier modes as eigenbasis of the graph Laplacian matrix. This treatment leads to the orthonormal eigenvectors, thus brings desirable algebraic proprieties. Irion and Saito (2015) pointed out that the spectrum of graph Laplacian can be misleading if treated as the frequency counterparts for general undirected graphs. They further developed two types of multiscale basis dictionaries on graphs, hierarchical graph Laplacian eigenbasis and generalized Haar-Walsh wavelet packet based on recursive graph partitionings. In a later study, the extended generalized Haar–Walsh transform was proposed to simultaneously consider the time domain and the frequency domain of the input graph signal (Saito and Shao, 2022).

One major limitation of the above studies is that the methods developed based on undirected graphs ignore the directed information in the system. The asymmetric feature is especially non-negligible for traffic flow in transportation networks. Several methods have been proposed to deal with the directness of edges. Singh et al. (2016) considered Jordan eigenvectors of directed Laplacian as graph harmonics and corresponding eigenvalues as graph frequencies. The natural frequency definition is maintained, meaning that the eigenvalues corresponding to eigenvectors with small variations are treated as low frequencies and vice versa. Deri and Moura (2016) defined the graph Fourier transform based on Jordan decomposition of adjacency matrix. However, several issues are still unsolved if using Jordan decomposition for adjacency matrix. One issue is that the resulting Fourier basis is not orthonormal. Also, the notion of frequency cannot be carried by the Fourier basis, meaning that low frequency does not necessarily represent slowly varied graph signals. To deal with the drawbacks of Jordan decomposition based graph Fourier transform, several studies

focused on other graph operators. Sevi et al. (2018) investigated the eigenfunctions based on the random walk matrix. Furutani et al. (2019) extended graph Laplacian to Hermitian Laplacian, where the orthonormality can be preserved. Optimization based approach are also used to design GSP for directed graphs. Sardellitti et al. (2017) imposed the orthonormality of the graph Fourier basis and formulated a non-convex optimization problem. They formulated the objective as the Lovász extension, a continuous extension of graph cut size, and orthogonality is controlled by constraints. Following similar intuition, Shafipour et al. (2018) formulated a constrained non-convex optimization to obtain the orthonormal Fourier basis such that corresponding frequencies are maximally spread over the entire spectral domain. However, due to the limitation of the nonconvex formulation, the uniqueness of the solution cannot be guaranteed.

Existing studies have demonstrated the capability of nonparametric regression and graph based filtering. However, challenges and opportunities remain. Most existing studies focus on undirected graphs, yielding well-behaved eigenfunctions of the graph matrices. Yet, this treatment would naturally ignore the asymmetric structure of the data, including traffic flows in the highway networks, sensor networks with directional information propagation such as wind or water flow measurements, network traffic and routing. For the same reason, undirected GSP might lead to the loss of crucial information about the data structure or the system dynamics. For instance, in social media networks, the 'following' relationships are often not mutual, and the direction of the relationship can provide valuable information. Furthermore, undirected graphs cannot represent causality in systems where the order of events or the direction of information flow matters, such as citation networks, web page networks, or dynamical systems. Therefore, there could be benefits in incorporating the directional structure into the estimation and filtering tasks.

In this study, we focus on univariate nonparametric regression for estimation on large-scale directed graphs. We approach the graph signal filtering problem under the synthesis framework by leveraging a Continuous-time Markov Chain (CTMC) based basis. The di-

rected combinatorial Laplacian can act as the negative generator matrix of the CTMC. The method can be easily applied to the cases where the graph structures are given as prior or to the general point cloud data by leveraging the KNN or proximity graphs.

The remainder of this chapter is organized as follows. In section 2, we introduce the Continuous-time Markov Chain (CTMC) based filtering on directed graphs. In section 3, we further discuss some characteristics of the proposed methods, including the kernel interpretation and computation acceleration. In section 4, we present the performance on both synthetic and real-world case studies. Section 5 offers conclusions and possible extensions.

## 4.2 CTMC Filtering on Directed Graphs

### 4.2.1 Directed Graphs and Continuous-time Markov Chain

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a directed graph, with vertices (nodes) $\mathcal{V} = v_1, \ldots, v_n$ and directed edges (links) $\mathcal{E} = e_1, \ldots, e_u$. The adjacency matrix $A$ captures the pairwise similarity among the nodes. In practice, the graph $\mathcal{G}$ can either be provided directly from domain applications or be derived from relational data or nearest-neighbor proximity. When a graph structure is not known as prior, one can construct a proximity graph or KNN graph from given data. One example of the proximity graph is defined as

$$A_{i,j} = \begin{cases} e^{-\|x_i - x_j\|_2^2/c} & \text{if } x_i \text{ and } x_j \text{ are close,} \\ 0 & \text{otherwise} \end{cases} \tag{4.2.1}$$

From the given graph structure, we have $d_i = \sum_{j=1}^{n} A_{ij}$. The degree matrix is then given as $D = diag(d_i)$ and the corresponding combinatorial graph Laplacian can be obtained as $L = D - A$. For the directed graph case, we use the out-degree $D := D_{out} = diag(\sum_j A_{ij})$. This would ensure the sum of each row of $L$ is zero, i.e. $\sum_{i,j \in \mathcal{N}} L_{ij} = 0$, as required for the generator matrix $Q$ for Continuous-time Markov Chain (CTMC) to be discussed below. Note that in-degree is equal to out-degree for undirected graphs, thus our methods are still applicable for undirected cases.

Next, let us shift our attention to CTMC. Consider a stochastic process $\{X(t), t \in [0,\infty)]\}$ with a countable state space $\mathcal{S} \subset \{0, 1, 2, \dots\}$. A CTMC consists of three components, a jump chain as a discrete-time Markov chain with the set of states and transition probabilities $p_{ij}$ and a set of holding time parameters $\tau_i$ that controls the amount of dwelling time at each state. If $X(t) = i$, the time until the state changes follows $Exponential(\tau_i)$ distribution. The probability of going from state $i$ to state $j$ is denoted as $p_{ij}$, defined as

$$\begin{aligned} P_{ij}(t) &= P(X(t+s) = j \mid X(s) = i) \\ &= P(X(t) = j \mid X(0) = \mathrm{i}), \quad \text{for all } s, t \in [0,\infty) \end{aligned} \tag{4.2.2}$$

and $P(0)$ equals to the identity matrix $P(0) = I_n$.

An effective way to understand CTMC is built on the concept of the generator matrix Q. The generator represents the transition rate from state $i$ to state $j$ as $g_{ij} = \tau_i p_{ij}$, with diagonal elements $g_{ii} = -\sum_{j \neq i} g_{ij} = -\tau i$. The sum of the rows in Q is zero, i.e. $\sum_j Q_{ij} = 0, \forall i$. Then we have the forward relationship

$$P_t' = P_t Q \tag{4.2.3}$$

and backward relationship

$$P_t' = Q P_t \tag{4.2.4}$$

The matrix-valued differential equation has a unique solution for the transition matrix function as

$$P_t = e^{tQ} = \sum_{n=0}^{\infty} \frac{(tQ)^n}{n!} \tag{4.2.5}$$

for all $t \geq 0$,

One may recognize that the infinitesimal generator $Q$ is the key bridge between CTMC and the graph Laplacian originated from algebraic graph theory. If we let the graph Laplacian be $L = -Q$, the directed graph $\mathcal{G}$ can be modeled as a CTMC, with state space $\mathcal{S}$ matching node space $\mathcal{N}$. Thus, we have the transition matrix for a directed graph $\mathcal{G}$ as the

exponentiation of the negative combinatorial graph Laplacian over time.

$$P_t = e^{tQ} = e^{-tL} = \sum_{n=0}^{\infty} \frac{(-tL)^n}{n!} = I + (-t)L + \frac{1}{2}t^2L^2 + (-\frac{1}{6}t^3)L^3 + \dots \tag{4.2.6}$$

for all $t \geq 0$.

There has been effort to use spectral connectivity analysis for kernel smoothing based on the random walk from discrete-time Markov chain (Lee and Wasserman, 2010). However, the discrete random walk is not directly applicable to the directed graphs, because of the fundamental assumption where the probability of transitioning from one node to another is equal in both directions if there is an edge connecting them. For a directed graph, this assumption generally does not hold, because an edge might only allow movement in one specific direction. Therefore, while discrete random walks provide a powerful tool for signal processing undirected graphs, their usefulness is limited when it comes to directed graphs for directed information flows.

### 4.2.2 CTMC Filtering on Directed Graphs

In this section, we present the CTMC filtering framework. Assume we observe node-based data $f \in \mathbb{R}^n$ defined based on the given directed graph. Consider the normal error model

$$f_i = r(x_i) + \epsilon_i, i = 1, \dots, N \tag{4.2.7}$$

with iid sub-Gaussian random error $\epsilon_i$ with parameter $\sigma$. The moment generating function of X is upper bounded by the moment generating function of a Gaussian random variable with mean 0 and variance $\sigma^2$ .

$$\mathbb{E}[e^{t\epsilon}] \leq e^{\frac{\sigma^2 t^2}{2}} \tag{4.2.8}$$

Our goal is to estimate the nonparametric regression function $r(\cdot)$ that produces a denoised version of observations $f$.

$$f = P_t^\top \beta + \epsilon \tag{4.2.9}$$

**CTMC Filtering:** $$\min_{\beta} \|f - P_t^T \beta\|_2^2 + \lambda \|\beta\|_2^2 \tag{4.2.10}$$

The first term measures the distortion between the observed signal $f$ and the estimated signal. The row stochastic matrix $P_t$ is used as the basis for the filtering operation. The $\ell_2$ regularized term is included to prevent overfitting and promote numerical stability. The gradient of the objective is given as

$$\frac{\partial h}{\partial \beta} = -2P_t f + 2P_t P_t^\top \beta + 2\lambda\beta \tag{4.2.11}$$

Thus we have the solution

$$\hat{\beta} = (P_t P_t^\top + \lambda I)^{-1} P_t f \tag{4.2.12}$$

Finally, we obtain the CTMC filtering estimate is

$$\begin{aligned} \hat{f} &= P_t^\top \hat{\beta} \\ &= P_t^\top (P_t P_t^\top + \lambda I)^{-1} P_t f \\ &= S_{\lambda,t} f \end{aligned} \tag{4.2.13}$$

with $S_{\lambda,t} = P_t^\top (P_t P_t^\top + \lambda I)^{-1} P_t$ as the linear smoother.

The probability distribution of the random walk at time $t$,given that it started at node i (state i) is given by the row vector $P_t(i,:) = Prob\{S(t)|S(0) = i\}$, the element $P_t(i,j)$ is the probability of transition from $i$ to $j$ within time $t$. This also matches the motivations from the semi-supervised smoothness assumption (Wasserman and Lafferty, 2007), which states as the regression function $m(x) = E[Y|X = x]$ is smooth where the density $p(x)$ of $X$ is large. $Y_i$ and $Y_j$ should be similar with high probability if there is a path connecting $X_i$ and $X_j$ on which $p(x)$ is large. In fact, in the next section, we will show the CTMC embedding can be viewed as a type of kernel ridge regression.

This design incorporates both local and higher-order neighborhood (multi-hop) information enabling heterogeneous degree of smoothness and varying levels of adaptive of the

estimated signal. When t is small, we have $P_t \approx I - tL$. The transition matrix depends on the local topology of the graph. When t is large, we have $P_t \approx \mathrm{e}^{-t\lambda_2}\phi_2\phi_2^{\mathrm{T}}$ with $\lambda_2$ being the smallest non-zero eigenvalue and $\phi_2$ being the eigenvector associated to the Laplacian. Thus the transition is governed by the global graph structure.

### 4.2.3 Selecting Parameters

We know that a Markov chain is irreducible if and only if its state graph is strongly connected. A chain is irreducible if one can go from any state to any other state in finite time. An irreducible chain is recurrent if the probability that we return to this state in finite time is one, i.e. $r_{ij} = Pr[T_j < \infty | X_o = i] = 1$. If a Markov chain is irreducible and positive recurrent, then it has a unique stationary distribution $\pi$.

That is, if the directed graph $\mathcal{G}$ is strongly connected, we could obtain a nice convergent behavior for the transition matrix $P_t$. A strongly connected graph is one where there is a directed path from any node to any other node. This is a common characteristic shared among transportation networks and sensor networks.

Therefore, we can achieve the stationary distribution for a large enough $t$ following the Theorem in Chapter 20 by Levin and Peres (2017), which we now restate here

**Theorem 2.** *For an irreducible CTMC, there exist a unique stationary distribution, such that $\pi P_t = \phi$ and the total variation distance is monotone non-increasing in t*

$$\max_{x\in\mathcal{X}} \|P_t(x,\cdot) - \pi\|_{\mathrm{TV}} \to 0 \quad as \quad t \to \infty \tag{4.2.14}$$

Note this theorem does not require P is aperiodic for CTMC.

The transition probability matrix P(t) is continuous for all $t \geq 0$. When the random walks become long (large $t$), the distribution of the data on vertices will converge to the stationary distribution. This would lead to a rank-1 projection basis in 4.2.10. In this study, assume we have $q$ observations of $f$, we choose to select parameters $t$ and $\lambda$ using cross-validation.

## 4.3 Properties and Extensions

### 4.3.1 Kernel Ridge Regression Interpretations

In this section, we show that the proposed CTMC filtering can be viewed as a type of kernel ridge regression (Zhang et al., 2013; Alaoui and Mahoney, 2015). First, recall the standard ridge regression for data $(X, y)$

$$\min \|y - Xw||_2^2 + \lambda \|w\|^2 \tag{4.3.1}$$

the solution can be as

$$\begin{aligned} \hat{w} &= (X^T X + \lambda I_D)^{-1} X^T y \\ &= X^T (X X^T + \lambda I_D)^{-1} y \end{aligned} \tag{4.3.2}$$

The last step is because

$$\left(X^T X + \lambda I\right) X^T = X^T X X^T + \lambda X^T = X^T \left(X X^T + \lambda I\right) \tag{4.3.3}$$

we can write this as

$$X^T = \left(X^T X + \lambda I\right)^{-1} X^T \left(X X^T + \lambda I\right) \tag{4.3.4}$$

thus we have

$$X^T \left(X X^T + \lambda I\right)^{-1} = \left(X^T X + \lambda I\right)^{-1} X^T \tag{4.3.5}$$

Now, let a set of dual variables be

$$\mu = (X X^T + \lambda I_N)^{-1} y \tag{4.3.6}$$

we can write the solution with dual variables as the linear combination of training data vectors

$$\hat{w} = X^T \mu = \sum_{n=1}^{N_D} \mu_n x_n \tag{4.3.7}$$

If we define the feature vector using $i$th row of transition matrix $P_t$ as $x_i \to \phi(x_i) = P_t(i)$. After replacing $X$ in standard ridge regression to the kernelized version as $\phi(x)$, the solution

to the kernel ridge form is identical to our solution to CTMC smoothing 4.2.10

$$\begin{aligned}\hat{\beta} &= (\phi(X)^T\phi(X) + \lambda I_D)^{-1}\phi(X)^T y \\ &= (P_t P_t^\top + \lambda I)^{-1} P_t f\end{aligned} \tag{4.3.8}$$

we can also write $\hat{\beta}$ as

$$\begin{aligned}\hat{\beta} &= P_t(P_t^\top P_t + \lambda I)^{-1} f \quad \text{matrix inversion lemma} \\ &= P_t \gamma\end{aligned} \tag{4.3.9}$$

with dual variable as $\gamma = (P_t^\top P_t + \lambda I)^{-1} f$.

if we have a new test data vector, the prediction can be obtained by the representer theorem,

$$f_{new} = \hat{\beta}^T \phi(x_{new}) = \sum_{n=1}^{N} \gamma_n K(x_n, x_{new}) = \kappa^T (K + \lambda I)^{-1} f \tag{4.3.10}$$

with kernel $K(x_i, x_j) = \phi(x_i)^T \phi(x_i)$ and $\kappa = [K(x_{new}, x_1), \ldots, K(x_{new}, x_N)]$ and $\gamma = (K + \lambda I)^{-1} f$

Now we have shown the CTMC smoothing is exactly a type of kernel ridge regression with CTMC transition matrix as the kernel.

### 4.3.2 Computation

We have shown that CTMC filtering as a linear smoother has a closed-form solution 4.2.13. The main computational challenge is that the transition matrix $P_t = e^{-tL}$ and the combinatorial Laplacian $L$ are not simultaneously diagonalizable, because $L$ is not normal for a general directed graph. Therefore, to select the tuning parameter $t$, one needs to enumerate the computationally expensive matrix exponential operations.

Here we introduce some ways to improve the computational efficiency. We start with the symmetric case, where the graph $\mathcal{G}$ has symmetric combinatorial Laplacian $L$. In practice, this can come from 1) undirected graphs naturally with symmetric graph Laplacian; or 2) directed graphs with symmetric pairwise relationship, such as friendship and relative networks.

For a symmetric graph, we can write the spectral decomposition of the Laplacian as $L = V\Lambda V^{-1}$, where $\Lambda$ is the diagonal matrix of distinct eigenvalues. Then the matrix exponential can be computed efficiently over a grid of $t$ as

$$\begin{aligned} P_t &= e^{-tL} \\ &= \sum_{n=0}^{\infty} \frac{(-tL)^n}{n!} \\ &= V \sum_{n=0}^{\infty} \frac{1}{n!} \begin{bmatrix} -\lambda_1^n & 0 & 0 & 0 \\ 0 & -\lambda_2^n & 0 & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & -\lambda_N^n \end{bmatrix} V^{-1} \\ &= V \begin{bmatrix} -e^{\lambda_1 t} & 0 & 0 & 0 \\ 0 & -e^{\lambda_2 t} & 0 & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & -e^{\lambda_N t} \end{bmatrix} V^{-1} \end{aligned} \tag{4.3.11}$$

For the general case, where $L$ and $P_t$ are asymmetric, we need to reply on other methods to efficiently compute the matrix exponential. Krylov subspace methods have been widely used in linear algebra and matrix operations (Saad, 1981; Watkins, 2007; Bai, 2015). Among them, the Arnoldi iteration finds an approximation to the matrices and eigensystem by constructing an orthonormal basis of the Krylov subspace (Arnoldi, 1951). Here we use Arnoldi iteration to approximate the matrix exponential following the design in Saad (1992).

The goal here is to efficiently approximate the action of a matrix exponential on a vector $P_t f$. For convenience of notation, we write $v := f$ as the initial vector. We first use the Arnoldi iteration to obtain an orthogonal basis $V_m$ as $\{v_1, v_2, \dots, v_m\}$, which is an orthonor-

mal basis of subspace $K_m$.

$$
\begin{aligned}
&1.\ \text{Compute } v_1 = v/\|v\| \\
&2.\ \text{For } j = 1, 2, \ldots, m \\
&\qquad w := Lv_j \\
&\qquad \text{For } i = 1, 2, \ldots, j \\
&\qquad\qquad h_{i,j} := (w, v_j) \\
&\qquad\qquad w := w - h_{i,j} v_j \\
&3.\ \text{Compute } h_{j+1,j} = \|w\|_2 \text{ and } v_{j+1} = w/h_{j+1,j}
\end{aligned}
\tag{4.3.12}
$$

As the result, we obtain $V_m := \{v_1, v_2, \cdots, v_m\}$ of dimensions $n \times m$ as an orthonormal basis of the Krylov subspace $K_m$ with $m \ll n$. Let $H_m := [h_{ij}]$ be the $m \times m$ upper Hessenberg matrix, by Arnoldi iteration, we have

$$LV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T \tag{4.3.13}$$

i.e. $v_{m+1}$ satisfies an (m+1)-term recurrence relation involving itself and the previous Krylov vectors.

$$Lv_m = h_{1m} v_1 + \cdots + h_{mm} v_m + h_{m+1,m} v_{m+1} \tag{4.3.14}$$

By the orthogonality of $V_m$ (as $V_m^\top V_m = I_m$, and $v_m^\top v_{m+1} = 0$), we have

$$H_m = V_m^T L V_m \tag{4.3.15}$$

which represents the projection of $L$ onto the Krylov subspace $K_m$, with respect to the basis $V_m$.

The relationship with the subsequent iterations

$$LV_m = V_{m+1} \tilde{H}_m \tag{4.3.16}$$

$\tilde{H}_m$ is given by appending $h_{m+1,m}$ to the last row of the upper Hessenberg matrix $H_m$.

$$H_m = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \cdots & h_{1,m} \\ h_{2,1} & h_{2,2} & h_{2,3} & \cdots & h_{2,m} \\ 0 & h_{3,2} & h_{3,3} & \cdots & h_{3,m} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_{m,m-1} & h_{m,m} \end{bmatrix} \tag{4.3.17}$$

Then the matrix is projected onto a smaller space, with the exp is applied to the reduced matrix.

$$e^L v \approx \beta V_m e^{H_m} e_1 \tag{4.3.18}$$

with $\beta = \|v\|_2$. The matrix exponential on $v$ can be approximated as ($\mathcal{O}(MN^2)$). The detailed theoretical analysis remains to be shown in future work.

## 4.4 Case Studies in Networked Data

In this section, we present two case studies of CTMC filtering on directed graphs, including a synthetic example based on Boston highway network and a real-world example based on the Los Angeles (LA) sensor network.

### 4.4.1 Synthetic examples on Boston Highway Network

We first present the performance of CTMC filtering on Boston network. The network has 74 nodes and 258 directed links, as shown in Figure 4.1. In the figure, we use color to show the number of connections of each node, i.e. symmetric version of node degree. Clearly, there exist three high-density hubs that are closely connected nearby nodes. These hubs are also inter-connected by several corridors among them. When constructing the edge weight $A$, we first simulate a set of traffic link flows following User Equilibrium traffic assignment (Sheffi, 1985). Then we use the link flow as the edge weights with $A_{ij}$ representing the number of vehicles traversing from node $i$ to node $j$. The assumption is that the signals on the nodes should be more closely connected when there is more traffic in between. Based on the constructed graph $\mathcal{G}$, we can obtain the directed combinatorial graph Laplacian $L$. Figure 4.2

shows the difference of the Laplacian for directed and its undirected version $L - 1/2(L + L^\top)$ with each row and column representing a node in set $\mathcal{V}$. This quantity provides information on how the incidence edges change when we transform undirected graphs to directed graphs, effectively measuring the degree of asymmetry of the graph $\mathcal{G}$.

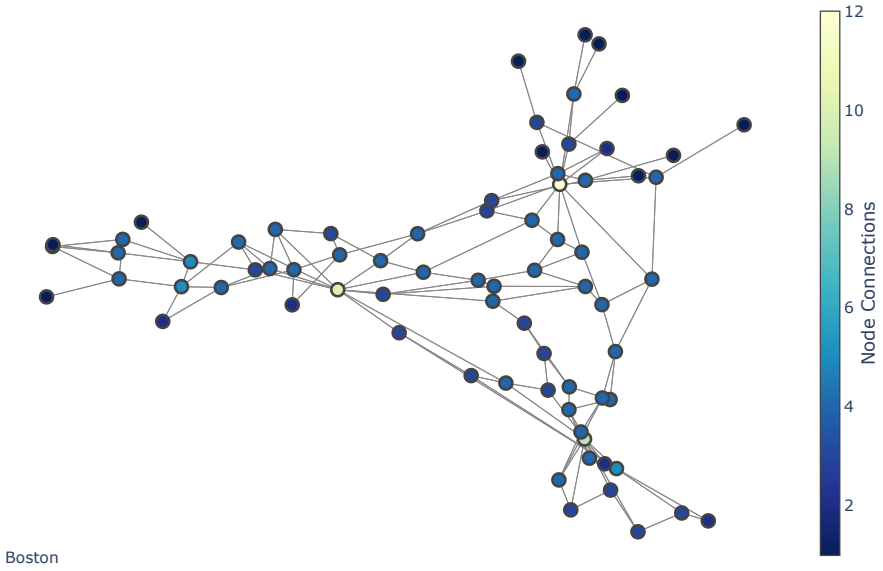



Figure 4.1: Boston Network

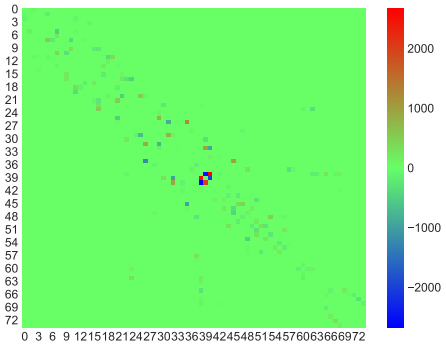



Figure 4.2: Asymmetric structure of L

Next, we use two different experiments to simulate the signal and noise on the graph. First, we present the experiment on a homogeneous continuous-time random walk. The

procedure is designed s follows: 1) simulate a set of underlying standard Gaussian $z_i$; 2) apply a CTMC transition $P_0$ to the standard Gaussian with random $t$; and 3) add iid Gaussian noise $\epsilon_i$ $\mathcal{N}(0, \sigma^2)$. These steps would generate a noisy observation of the graph signal $f \in \mathbb{R}^n$. These steps are repeated to generate total $q$ samples for the noisy signals $F \in \mathbb{R}^{n \times q}$, which is further separated to cross-validation set and evaluation set. One example of the 'true' noiseless signal is shown in Figure.4.3.

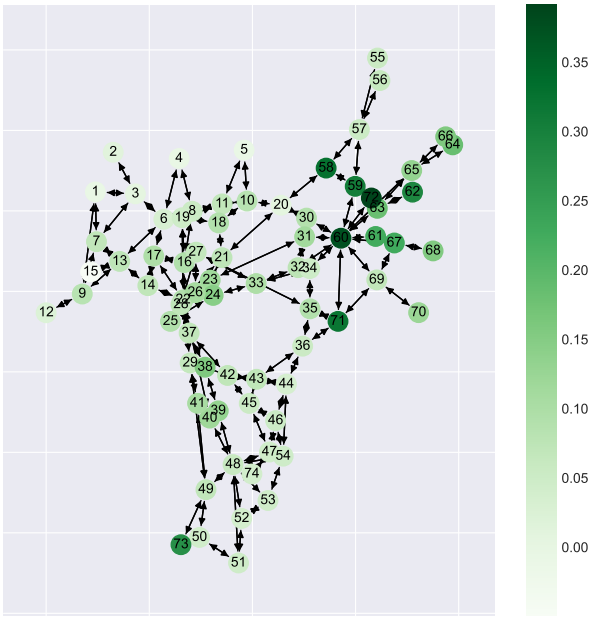



Figure 4.3: One example of underlying true graph signal

For the second experiment, we apply N-Random Walks onto the graph, following the procedure: 1) randomly select a node in the graph; 2) run discrete-time random walk with random step length $\rho \in \mathcal{N}^+$; 3) apply indicator function to the nodes visited i.e. the graph signal of a node is the number of times the node is visited by this random walk; and 4) repeat the steps from 1 to 3 and obtain one observation of graph signal vector $f \in \ltimes$ as the summation of multiple random walk indicator functions. At last, similarly, the procedure is repeated for $q$ times, producing the noisy data matrix $F \in \mathbb{R}^{n \times q}$.

For comparison purpose, we implement the Laplacian smoothing in 4.1.2, which is ded-

icated to undirected graphs by design. Besides, we also compare the results with a spatial smoothing technique, Gaussian Kernel Smoothing.

$$f(x) = \frac{1}{N}\sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right) \tag{4.4.1}$$

where the Gaussian kernel is defined as

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} \tag{4.4.2}$$

In order to apply these two methods, we need to convert the directed graphs to its undirected version. We take every directed edge and replace it with two edges with opposite orientation of half of the original weight. This is the typical treatment in practice when one can transform directed graphs to undirected graphs, and apply undirected 'vanilla' GSP and filtering methods.

Next, we present results with varying negative SnR to demonstrate the performance of CTMC Filtering, defined as $10log_{10}(n\sigma^2/\|f\|_2^2)$ with $\sigma^2$ being the variance of the noise. Our method generates the smallest log MSE compared to the other two methods.

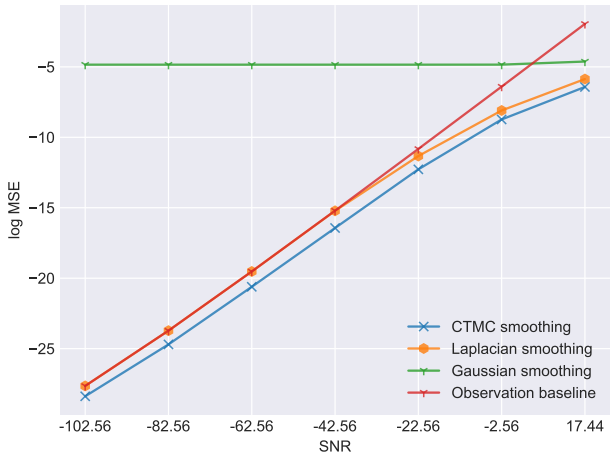



Figure 4.4: Homogeneous Random Walk on Boston Network

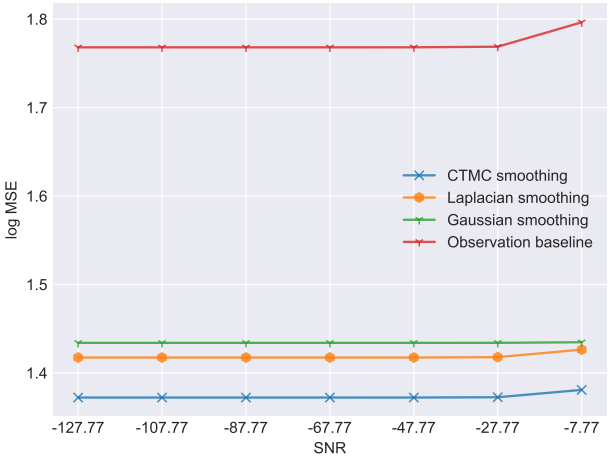



Figure 4.5: N Random Walks on Boston Network

### 4.4.2 Real world example - Los Angeles Sensor Network

Next, we present a case study based on real-world data of the traffic flow sensor measurements in the Los Angeles County. The sensor data is retrieved from Performance Measurement System (PeMS), an online system developed and maintained by the California Department of Transportation (Caltrans)[1]. These detectors provide information on traffic volume, vehicle classification, speed, and occupancy, and the system also collects and displays incident and lane closure information. We use the traffic flow data of the first two weeks in October 2022 for LA county region. For this study, we focus on mainline sensors, totaling 1906 sensors in the region. Here we show the example sensor readings on traffic flow for morning peak, afternoon peak and evening off-peak on October 3rd 2022, in Figure 4.6, 4.7 and 4.8 respectively. The color in the figure represent the aggregated total traffic flow in the 5-min interval. For example, Figure 4.6 shows the total flow that passes through each sensor from 9:00 a.m. to 9:05 a.m.

[1] $https://pems.dot.ca.gov/$

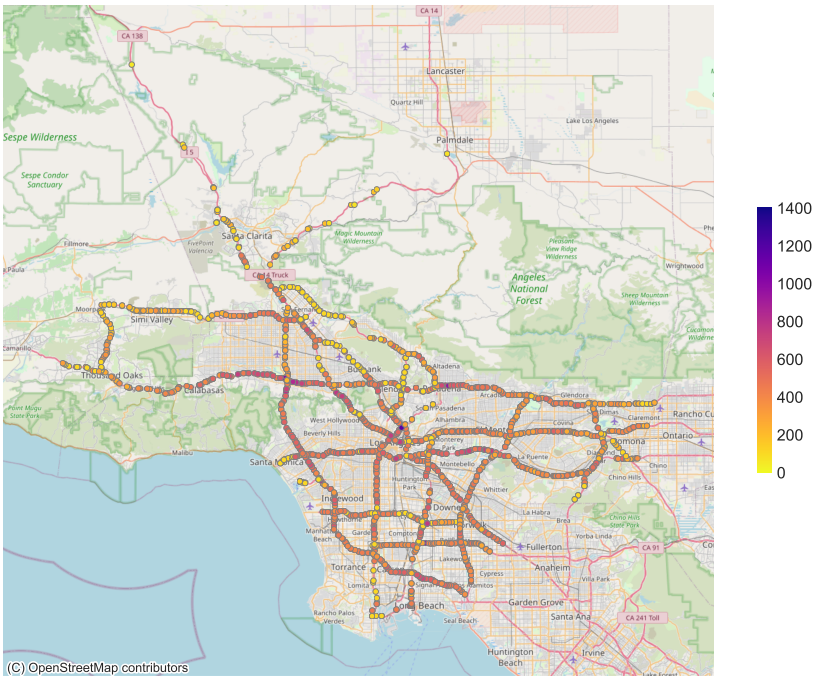



Figure 4.6: Sensor flow in Los Angeles County in the morning (9 a.m.)

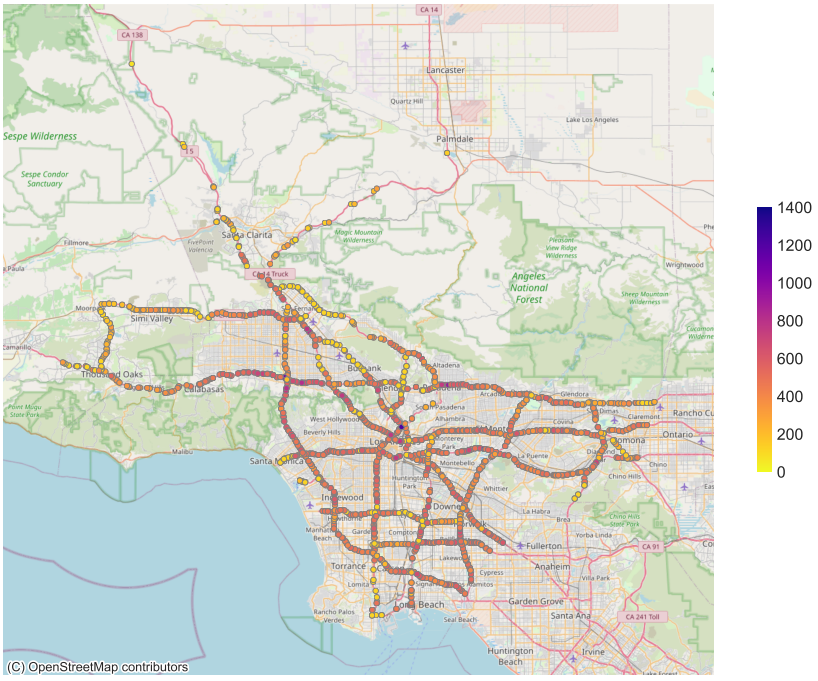



Figure 4.7: Sensor flow in Los Angeles County in the afternoon (5 p.m.)

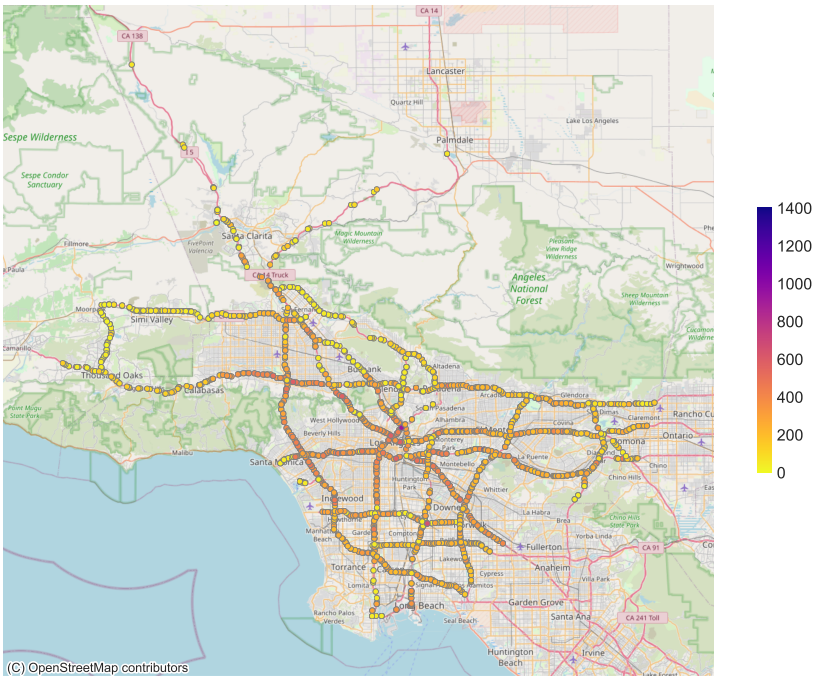



Figure 4.8: Sensor flow in Los Angeles County in the evening (10 p.m.)

In this case study, we focus on the sensor network for the detectors deployed throughout the region, and treat sensor readings as the graph signal $f$. Different from the first experiment, we do not directly use the underlying road network where the sensors are deployed for several considerations. First, PeMS system doesn't have sensors deployed on all road sections, i.e. links in the context of highway graph representation. This results in an incomplete set of observations, where only a subset of links within the graph provide signal observations. This particular characteristic contradicts our setting, where we assume the availability of a complete set of observations. Second, in the road network setting, the positioning of sensors on road segments would produce link-based data. This is different from our node-based signal-noise model. One alternative is to leverage the link-node incidence matrix to transform the observations to the node space. However, the noise model will be impacted due to the linear transformation. For those reasons, we directly work with sensor detection and corresponding sensor networks. We recognize the opportunities in incorporat-

ing the underlying physical road network and leave the incorporation of the physical road network to future research.

The asymmetric structure for transportation sensor network comes from several sources. The first factor contributing to the asymmetry structure is the irregular spatial distribution of the sensors. Specifically, for a node $n_i$ that falls within the K-neighborhood region $\mathcal{N}_j$ of another node $n_j$, it's not guaranteed that the neighborhood $\mathcal{N}_i$ of $n_i$ will include $n_j$. This is due to varying densities of sensor placement throughout the spatial region. Second, on the same route, node (sensor) pairs installed on upstream-downstream vs. downstream-upstream can have different local correlation since the shockwave and congestion propagation of the traffic flow can be asymmetric for different directions. Furthermore, the travel time faced by a driver is heavily dependent on the traffic condition in front of them (downstream) and very little on the traffic condition behind them (upstream), known as the anisotropic property. The third cause for asymmetry arises from the sensors located on different routes or different directions. Depending on the relative directions of the routes - whether they're opposite or more parallel to each other - the relationships between the sensors can differ significantly. Besides, there may be strongly directional origin-destination travel demand during peak hours, such as morning commute period where travelers drive from residential areas to business areas. While the asymmetric travel demand might be less prominent during off-peak periods.

We construct a directed KNN-based graph for sensor networks that leverage the local structure of flow information. First, we compute the pairwise hasrsine distance for all nodes in the region as $\mathcal{D} \in \mathbb{R}^{n \times n}$. Next, based on the relationship between the spatial positioning of the sensor network and the underlying highway network, we make the following adjustment to the distance $\mathcal{D}$ based on several heuristics: 1) if two sensors $n_i$ and $n_j$ are on the same highway route with the same direction, the distance is discounted by $\alpha_1$ as $\tilde{\mathcal{D}}_{ij} = \mathcal{D}_{ij}/\alpha_1$; 2) if two sensors $n_i$ and $n_j$ are on the same route but on opposite directions, the distance is increased as $\tilde{\mathcal{D}}_{ij} = \mathcal{D}_{ij} \times \alpha_2$; 3) if two sensors are on different routes and on different directions,

we also increase the distance as $\tilde{\mathcal{D}}_{ij} = \mathcal{D}_{ij} \times \alpha 3$. In the LA example, we let $\alpha_1 = 5$, $\alpha_2 = 10$ and $\alpha_3 = 2$. Then, we construct a KNN graph based on the adjusted distance $\tilde{\mathcal{D}}$. Further, a Gaussian RBF kernel is applied to $\mathcal{D}$ to translate the distance for similarity $A$. Finally, the combinatorial Laplacian $L$ and the transition matrix $P_t$ are computed based on the similarity matrix $A$.

We show the asymmetric structure for $L - 1/2(L + L^\top)$, as in Figure 4.9. Incorporating these asymmetries through directed graphs allows us to capture more accurate and nuanced relationships between nodes, enhancing the performance of graph-based filtering techniques.

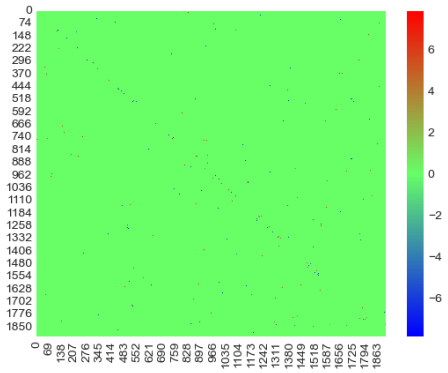



Figure 4.9: Asymmetric Structure

For this experiment, we apply a rolling aggregation window $\mathcal{T}$, ranges from 5 minutes to 1 hour to mimic different signal to noise ratio. That is, we aggregate the original 5-minute observations $f_t$ to $f' = \sum_{t \in \mathcal{T}} f_t$, with $\tau \in \{5, 10, 15, 20, 30, 60\}$. We assume the larger the aggregation window is, the smaller the noise level. This is because the aggregation window acts like a moving average low-pass filter.

The performance of our CTMC filtering is again compared with the Laplacian smoothing 4.1.2 and spatial kernel smoothing 4.4.1. After the filtering, we can further break down the aggregation window, obtaining the original 5-minute level unit MSE and RMSE.

In this case, due to the low noise level in the real world data, we show the MSE improve-

ment over baseline as raw observations, defined as $\Delta MSE = MSE_{baseline} - MSE_{CTMC}$. The MSE and unit MSE improvement results for the rolling window evaluation are shown in Figure 4.10 and 4.11. The RMSE and unit RMSE improvements results are shown in Figure 4.12 and 4.13. Among the methods tested, our CTMC filtering generates the best results.

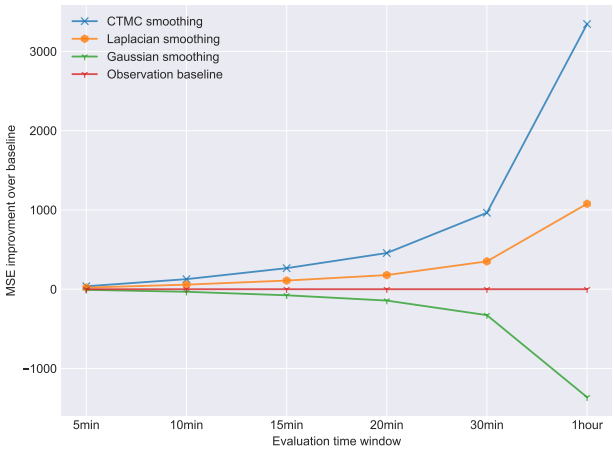



Figure 4.10: MSE improvement over baseline

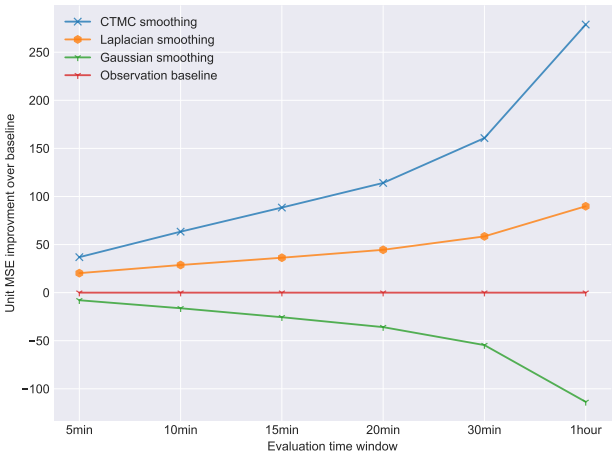



Figure 4.11: Unit MSE improvement over baseline

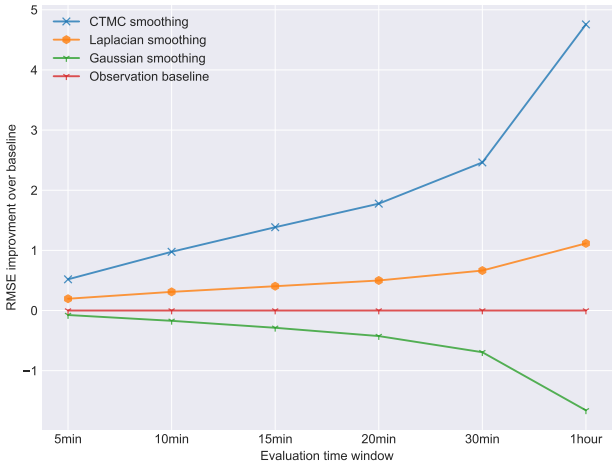



Figure 4.12: RMSE improvement over baseline

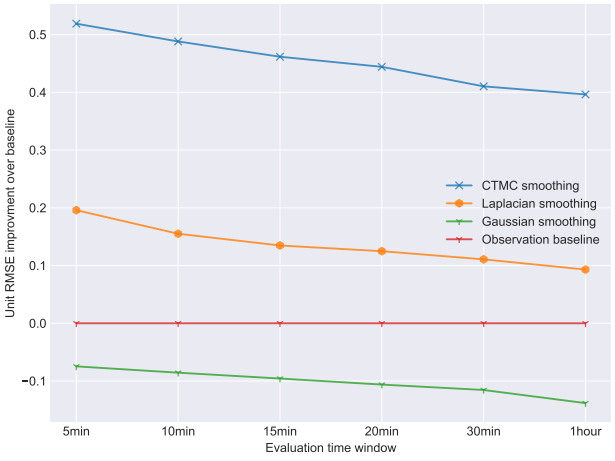



Figure 4.13: Unit RMSE improvement over baseline

## 4.5 Conclusion and Discussion

In this work, we propose a novel CTMC filtering approach for estimation on directed graphs. Built on the nonparametric regression, we bridge the stochastic process and graph theory through the directed combinatorial graph Laplacian and transition matrix in CTMC. Compared to filtering methods based on spatial information and undirected graph structure, our method is capable of capturing local asymmetric structure in data observations. We demonstrate the performance through synthetic and real-world case studies, showing the potential to incorporate heterogeneous structure in data.

For next steps, we will explore the conditions where directed graphs provide the most benefit compared to the undirected counterpart. For instance, in scenarios where the underlying graph is inherently undirected or where global trends are more relevant than local variations, undirected Laplacian smoothing or spatial smoothing might still be a reasonably simplified choice.

# Chapter 5

# Conclusions

With the rapid advancement of information technology and the surge in data availability, transportation systems have evolved into complex, large-scale networks characterized by complex dynamics and interactions. The information era generates vast amounts of diverse data, including traffic flows, vehicle trajectories, and travel behavior information, which present unprecedented opportunities for optimization and machine learning applications.

This dissertation work fills some important research gaps in understanding high-dimensional networked data in transportation systems. We have demonstrated the opportunities and potential challenges that come with applying optimization and statistical machine learning to high-dimensional networked data in transportation systems. The proposed methodologies have shown promising results in improving the efficiency, resilience, and sustainability of these systems.

In Chapter 2, we have developed a method for estimating OD demand at a network level by utilizing stochastic programming and multiple sets of observed link flow data. This methodology enables the simultaneous estimation of anticipated demand and the reconstruction of OD trips throughout observed time intervals. A novel aspect of our approach lies in viewing the problem through the lens of a two-stage stochastic programming framework. In this setting, demand parameter estimation is considered the first-stage decision, while

demand reconstruction is treated as a recourse decision that depends on various scenarios. This new perspective allows us to draw parallels between our estimation technique and the well-established field of stochastic programming, leading to several noteworthy advantages. Primarily, this systemic approach unlocks a wealth of pre-existing knowledge from the stochastic programming community, providing opportunities to leverage established solution methods for large-scale high-dimensional problems and modeling options for integrating additional risk preferences. We demonstrate the computational benefits of this approach by implementing a scenario decomposition method designed to address computational challenges posed by extensive scenarios.

In Chapter 3, we have proposed a novel representation learning framework that is specifically designed to incorporate the needs of downstream applications. We highlight the importance of incorporating application-driven knowledge in the loss function, especially when the application metric exhibits a different pattern compared to the raw data. This is a common occurrence in high-dimensional transportation network problems, where data elements interact with each other across a complex network structure. This extension to the body of literature on data-driven methods in transportation science addresses the growing demand for end-to-end data analytics, in line with our focus on high-dimensional networked data and machine learning. We demonstrated the capability of proposed methods through mobility demand estimation, emission estimation and anomaly detection tasks. The philosophy embedded in this approach could be extended beyond representation learning, finding applicability in data clustering, classification, and other related tasks. Our numerical results show promising indications of the existence of sparsity and dominant features in the traffic data, illustrating the potential effectiveness of manifold optimization techniques.

In Chapter 4, we have presented a CTMC based filtering framework on directed graphs. This novel method builds on nonparametric regression methods, via utilizing key components such as the directed combinatorial graph Laplacian and transition matrix. The method is capable of capturing heterogeneity in local adaptivity and asymmetries. CTMC filtering

provides an important advancement over traditional filtering methods, which typically only rely on spatial information or are limited to undirected graphs. As we have demonstrated in the case studies, these standard techniques, while robust in certain settings, often fall short when faced with the complex, heterogeneous, and asymmetric structures commonly found in data from large-scale networks, such as transportation networks. On the contrary, CTMC filtering is specifically designed to handle these complexities. It effectively leverages the structural information embedded in directed graphs, enabling it to capture the nuanced local and asymmetric features that are prevalent in real-world data observations. Our proposed CTMC filtering method represents a step forward in the field of estimation on directed graphs, providing a more powerful tool that can handle the heterogeneous and irregular structures of real-world data. As we continue to explore and refine this method, we believe that it will open up new possibilities to exploit the structure of high-dimensional networked data in various applications.

# Chapter 6

# Future Research Opportunities

In this dissertation we have developed tools in stochastic programming, representation learning and nonparametric regression to explore for mobility and networked data in transportation systems. This opens up several avenues for future research.

**1. Estimation and Decision Making with Heterogeneous Information**

Most of the topics we have explored thus far predominantly operate under the assumption of homogeneous data types, including networked traffic flow (OD/path/link flow), vehicle trajectory, as well as emission rates.

There could be opportunities to improve estimation quality if we can expand the scope to incorporate multiple data sources. These include mobile sensors enabled by connected and automated vehicles (CAV), image and video data collected by roadside units or mobile camera, and human behavior information on travel demand.

Representation learning and data fusion tools could help reveal the underlying structure of the data based on heterogeneous media observation for the same object. These approaches allow us to effectively extract and leverage meaningful information from a rich array of data sources, providing a more robust and complete representation of the observed phenomena.

Nonetheless, several challenges and uncertainties require more careful considerations:

- Data Quality and Confidence Levels: When multiple data sources are integrated, the

quality and confidence level of the information can vary significantly from one source to another. The question then arises: how do we determine the reliability of each data source and decide which to trust most? Addressing this issue requires developing sophisticated methods for assessing and comparing data quality across diverse sources.

- Spatial and Temporal Discrepancies: There might be discrepancies in the spatial or temporal information obtained from different sources. For example, congestion observed on the road segment $e_i$ at the time $t$ could be caused by an incident captured elsewhere on the segment $e_j$ or a surge in demand that occurred much earlier than $t$, or both effect combined. Understanding and accounting for these discrepancies is crucial in order to ensure accurate interpretation and meaningful utilization of the data.

**2. Information Acquisition, Partial Observations, and Semi-supervised Learning**

The second future direction is on partial observations. This happens everywhere in transportation systems. For example, the loop sensors and cameras are only installed on selected locations in the network. Demand survey can only cover limited population or market segmentation. Certain locations in the region have been neglected for many planning and operation decision makings due to lack of data, outdated physical infrastructure, or limited software support and usage, especially for underserved or disadvantaged communities.

Opportunities exist in the estimation of unobserved information through inference on passively collected data or through active information acquisition. For example, CTMC filtering we developed can be extended to semi-supervised learning setting if we incorporate harmonic functions on the graph that measures the information propagation in the networks. Also, one can actively route CAVs on the road to collect critical information in under-observed areas. This strategy could potentially alleviate issues arising from low spatial-temporal sampling rates by enhancing the richness and diversity of the data collected.

Therefore, several questions are worth considering:

- Critical Information: What constitutes critical information that could significantly

enhance our understanding of the current state or anticipated state of the system?

- Information Acquisition: What is the most effective and cost-effective way to acquire this critical information? How can we optimally balance the trade-off between the cost of information acquisition and the potential value of the data collected? How can we prioritize the collection of this information to maximize its utility in decision-making processes?

**3. Applications to Operation, Sustainability, and Equity**

While this dissertation primarily focuses on the methodology aspects, the developed frameworks also open the door to several practical applications, including

- Operation: These techniques allow for the creation of more robust and efficient transportation systems. For instance, OD estimation can be used to better understand the travel patterns within a transportation network, which in turn can inform the efficient allocation of resources and optimization of route planning. Furthermore, representation learning can reveal patterns and relationships in the data that were previously unrecognizable, thus improving the performance of predictive models used for operation management and control. Similarly, CTMC filtering can assist in real-time traffic prediction and monitoring, facilitating the quick detection of congestion, incidents, and service disruptions, and enabling rapid, effective responses.

- Sustainability: By understanding and predicting travel patterns and network dynamics, these methods can inform strategies to reduce traffic congestion, thereby lowering vehicle emissions and contributing to environmental sustainability. Furthermore, the insights gained from these techniques can guide the planning and development of more sustainable transportation options, such as public transit systems or cycling infrastructure.

- Equity: These techniques can also play a significant role in promoting equity in transportation systems. OD demand estimation, for instance, can reveal discrepancies in the use of transit services across different communities, highlighting areas where service improvements are mostly needed. Similarly, representation learning can identify patterns of inequality in transportation systems, such as the disproportionate impacts of traffic congestion or poor public transit services on disadvantaged communities. This can inform the development of policies aimed at improving transportation equity.

# Bibliography


Absil, P.-A., Mahony, R., and Sepulchre, R. (2009). *Optimization algorithms on matrix manifolds*. Princeton University Press.

Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer.

Agrawal, S. and Agrawal, J. (2015). Survey on anomaly detection using data mining techniques. *Procedia Computer Science*, 60:708–713.

Ahmed, S. (2004). *Mean-risk objectives in stochastic programming*. Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, Institut für Mathematik.

Alaoui, A. and Mahoney, M. W. (2015). Fast randomized kernel ridge regression with statistical guarantees. *Advances in neural information processing systems*, 28.

Arnoldi, W. E. (1951). The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of applied mathematics*, 9(1):17–29.

Artstein, Z. and Wets, R. J. (1995). Consistency of minimizers and the slln for stochastic programs. *J. Convex Anal*, 2(1-2):1–17.

Asif, M. T., Mitrovic, N., Dauwels, J., and Jaillet, P. (2016). Matrix and tensor based methods for missing data estimation in large traffic networks. *IEEE Transactions on intelligent transportation systems*, 17(7):1816–1825.

Attouch, H. and Wets, R. J.-B. (1994). *Epigraphical processes: laws of large numbers for random lsc functions*. Dép. des Sciences Mathématiques Paris.

Azaron, A., Brown, K., Tarim, S. A., and Modarres, M. (2008). A multi-objective stochastic programming approach for supply chain design considering risk. *International Journal of Production Economics*, 116(1):129–138.

Bai, Z.-Z. (2015). Motivations and realizations of krylov subspace methods for large sparse linear systems. *Journal of Computational and Applied Mathematics*, 283:71–78.

Barth, M., An, F., Younglove, T., Scora, G., Levine, C., Ross, M., and Wenzel, T. (2000). The development of a comprehensive modal emissions model. *NCHRP Web-only document*, 122:25–11.

Belkin, M. and Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pages 585–591.

Ben-Akiva, M., Macke, P. P., and Hsu, P. S. (1985). *Alternative methods to estimate route-level trip tables and expand on-board surveys*. Number 1037.

Birge, J. R. and Louveaux, F. (2011). *Introduction to stochastic programming*. Springer Science & Business Media.

Bureau of Public Roads (1964). Traffic assignment manual. Technical report, U.S. Dept. of Commerce, Urban Planning Division, Washington, D.C.

Cai, B., Xu, X., Jia, K., Qing, C., and Tao, D. (2016). Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198.

Carøe, C. C. and Schultz, R. (1999). Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24(1-2):37–45.

Cascetta, E. (1984). Estimation of trip matrices from traffic counts and survey data: a generalized least squares estimator. *Transportation Research Part B: Methodological*, 18(4):289–299.

Chambolle, A. (2004). An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20:89–97.

Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58.

Chen, S. S., Donoho, D. L., and Saunders, M. A. (2001). Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159.

Chriqui, C. and Robillard, P. (1975). Common bus lines. *Transportation science*, 9(2):115–121.

Chung, F. R. (1997). *Spectral graph theory*, volume 92. American Mathematical Soc.

Clifton, D. A., Hugueny, S., and Tarassenko, L. (2011). Novelty detection with multivariate extreme value statistics. *Journal of signal processing systems*, 65(3):371–389.

Collado, R. A., Papp, D., and Ruszczyński, A. (2012). Scenario decomposition of risk-averse multistage stochastic programming problems. *Annals of Operations Research*, 200(1):147–170.

Cui, P., Wang, X., Pei, J., and Zhu, W. (2018). A survey on network embedding. *IEEE transactions on knowledge and data engineering*, 31(5):833–852.

Cuzzocrea, A. (2019). Management and analytics of big data sources in intelligent smart environments: Where we are and where we are going. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 567–572.

Daganzo, C. F. and Sheffi, Y. (1977). On stochastic models of traffic assignment. *Transportation science*, 11(3):253–274.

De Cea, J. and Fernández, E. (1993). Transit assignment for congested public transport systems: an equilibrium model. *Transportation science*, 27(2):133–147.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29.

Deri, J. A. and Moura, J. M. (2016). New york city taxi analysis with graph signal processing. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1275–1279. IEEE.

Djukic, T., Flötteröd, G., Van Lint, H., and Hoogendoorn, S. (2012). Efficient real time od matrix estimation based on principal component analysis. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 115–121. IEEE.

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87.

Donoho, D. L. and Johnstone, I. M. (1995). Adapting to unknown smoothness via wavelet shrinkage. *Journal of the american statistical association*, 90(432):1200–1224.

Eichhorn, A. and Römisch, W. (2005). Polyhedral risk measures in stochastic programming. *SIAM Journal on Optimization*, 16(1):69–95.

Elad, M., Milanfar, P., and Rubinstein, R. (2007). Analysis versus synthesis in signal priors. *Inverse problems*, 23(3):947.

Fan, Y. and Liu, C. (2010). Solving stochastic transportation network protection problems using the progressive hedging-based method. *Networks and Spatial Economics*, 10(2):193–208.

Furutani, S., Shibahara, T., Akiyama, M., Hato, K., and Aida, M. (2019). Graph signal processing for directed graphs based on the hermitian laplacian. In *Proceedings of ECML-PKDD*.

Gipps, P. G. (1981). A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, 15(2):105–111.

Godsil, C. and Royle, G. F. (2001). *Algebraic graph theory*, volume 207. Springer Science & Business Media.

Goulart, J. d. M., Kibangou, A., and Favier, G. (2017). Traffic data imputation via tensor completion based on soft thresholding of tucker core. *Transportation Research Part C: Emerging Technologies*, 85:348–362.

Györfi, L., Kohler, M., Krzyzak, A., Walk, H., et al. (2002). *A distribution-free theory of nonparametric regression*, volume 1. Springer.

Hautamaki, V., Karkkainen, I., and Franti, P. (2004). Outlier detection using k-nearest neighbour graph. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 430–433. IEEE.

Hazelton, M. L. (2000). Estimation of origin–destination matrices from link flows on uncongested networks. *Transportation Research Part B: Methodological*, 34(7):549–566.

Hazelton, M. L. (2008). Statistical inference for time varying origin–destination matrices. *Transportation Research Part B: Methodological*, 42(6):542–552.

Hazelton, M. L. (2010). Statistical inference for transit system origin-destination matrices. *Technometrics*, 52(2):221–230.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.

Hoffmann, H. (2007). Kernel pca for novelty detection. *Pattern recognition*, 40(3):863–874.

Huang, G. and Loucks, D. P. (2000). An inexact two-stage stochastic programming model for water resources management under uncertainty. *Civil Engineering Systems*, 17(2):95–118.

Hvattum, L. M. and Løkketangen, A. (2009). Using scenario trees and progressive hedging for stochastic inventory routing problems. *Journal of Heuristics*, 15(6):527–557.

Irion, J. and Saito, N. (2015). Applied and computational harmonic analysis on graphs and networks. In *Wavelets and Sparsity XVI*, volume 9597, pages 336–350. SPIE.

Jebara, T., Wang, J., and Chang, S.-F. (2009). Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 441–448.

Ji, Y., Mishalani, R. G., and McCord, M. R. (2014). Estimating transit route od flow matrices from apc data on multiple bus trips using the ipf method with an iteratively improved base: method and empirical evaluation. *Journal of Transportation Engineering*, 140(5):04014008.

Jiang, B. and Dai, Y.-H. (2015). A framework of constraint preserving update schemes for optimization on stiefel manifold. *Mathematical Programming*, 153(2):535–575.

Johansen, S. (1980). The welch-james approximation to the distribution of the residual sum of squares in a weighted linear regression. *Biometrika*, 67(1):85–92.

Kataoka, S. (1963). A stochastic programming model. *Econometrica: Journal of the Econometric Society*, pages 181–196.

Ke, J., Zhang, S., Yang, H., and Chen, X. (2019). Pca-based missing information imputation for real-time crash likelihood prediction under imbalanced data. *Transportmetrica A: transport science*, 15(2):872–895.

Keogh, E., Lonardi, S., and Ratanamahatana, C. A. (2004). Towards parameter-free data mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 206–215.

King, A. J. and Rockafellar, R. T. (1993). Asymptotic theory for solutions in statistical estimation and stochastic programming. *Mathematics of Operations Research*, 18(1):148–162.

King, A. J. and Wets, R. J. (1991). Epi-consistency of convex stochastic programs. *Stochastics and Stochastic Reports*, 34(1-2):83–92.

Korf, L. A. and Wets, R. J.-B. (2001). Random lsc functions: an ergodic theorem. *Mathematics of Operations Research*, 26(2):421–445.

Lee, A. B. and Wasserman, L. (2010). Spectral connectivity analysis. *Journal of the American Statistical Association*, 105(491):1241–1255.

Levin, D. A. and Peres, Y. (2017). *Markov chains and mixing times*, volume 107. American Mathematical Soc.

Li, C. and Grossmann, I. E. (2021). A review of stochastic programming methods for optimization of process systems under uncertainty. *Frontiers in Chemical Engineering*, 2:622241.

Liu, C., Fan, Y., and Ordóñez, F. (2009). A two-stage stochastic programming model for transportation network protection. *Computers & Operations Research*, 36(5):1582–1590.

Liu, W., Li, X., Liu, T., and Liu, B. (2019). Approximating betweenness centrality to identify key nodes in a weighted urban complex transportation network. *Journal of Advanced Transportation*, 2019.

Lo, H., Zhang, N., and Lam, W. H. (1996). Estimation of an origin-destination matrix with random link choice proportions: a statistical approach. *Transportation Research Part B: Methodological*, 30(4):309–324.

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.

Maier, M., Luxburg, U., and Hein, M. (2008). Influence of graph construction on graph-based clustering measures. *Advances in neural information processing systems*, 21.

Marsland, S., Shapiro, J., and Nehmzow, U. (2002). A self-organising network that grows when required. *Neural networks*, 15(8-9):1041–1058.

McCord, M., Mishalani, R., Goel, P., and Strohl, B. (2010). Iterative proportional fitting procedure to determine bus route passenger origin-destination flows. *Transportation Research Record: Journal of the Transportation Research Board*, (2145):59–65.

Mulvey, J. M. and Vladimirou, H. (1991). Applying the progressive hedging algorithm to stochastic generalized networks. *Annals of Operations Research*, 31(1):399–424.

Newell, G. F. (1961). Nonlinear effects in the dynamics of car following. *Operations research*, 9(2):209–229.

Nguyen, S., Morello, E., and Pallottino, S. (1988). Discrete time dynamic estimation model for passenger origin/destination matrices on transit networks. *Transportation Research Part B: Methodological*, 22(4):251–260.

Noyan, N. (2012). Risk-averse two-stage stochastic programming with an application to disaster management. *Computers & Operations Research*, 39(3):541–559.

Ortega, A., Frossard, P., Kovačević, J., Moura, J. M., and Vandergheynst, P. (2018). Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828.

Pflug, G. C. (2003). Stochastic optimization and statistical inference. *Handbooks in operations research and management science*, 10:427–482.

Pimentel, M. A., Clifton, D. A., Clifton, L., and Tarassenko, L. (2014). A review of novelty detection. *Signal processing*, 99:215–249.

Ran, B., Tan, H., Feng, J., Wang, W., Cheng, Y., and Jin, P. (2016). Estimating missing traffic volume using low multilinear rank tensor completion. *Journal of Intelligent Transportation Systems*, 20(2):152–161.

Rockafellar, R. T. (1970). *Convex analysis*, volume 18. Princeton university press.

Rockafellar, R. T. (2018). Solving stochastic programming problems with risk measures by progressive hedging. *Set-Valued and Variational Analysis*, 26(4):759–768.

Rockafellar, R. T. and Wets, R. J.-B. (1991). Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of operations research*, 16(1):119–147.

Rockafellar, R. T. and Wets, R. J.-B. (2009). *Variational analysis*, volume 317. Springer Science & Business Media.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326.

Ruszczyński, A. (1997). Decomposition methods in stochastic programming. *Mathematical programming*, 79(1):333–353.

Ruszczyński, A. and Shapiro, A. (2003). Stochastic programming models. *Handbooks in operations research and management science*, 10:1–64.

Saad, Y. (1981). Krylov subspace methods for solving large unsymmetric linear systems. *Mathematics of computation*, 37(155):105–126.

Saad, Y. (1992). Analysis of some krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis*, 29(1):209–228.

Saffari, E., Yildirimoglu, M., and Hickman, M. (2020). A methodology for identifying critical links and estimating macroscopic fundamental diagram in large-scale urban networks. *Transportation Research Part C: Emerging Technologies*, 119:102743.

Saito, N. and Shao, Y. (2022). eghwt: The extended generalized haar–walsh transform. *Journal of Mathematical Imaging and Vision*, 64(3):261–283.

Santoso, T., Ahmed, S., Goetschalckx, M., and Shapiro, A. (2005). A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research*, 167(1):96–115.

Sardellitti, S., Barbarossa, S., and Di Lorenzo, P. (2017). On the graph fourier transform for directed graphs. *IEEE Journal of Selected Topics in Signal Processing*, 11(6):796–811.

Schultz, R. and Tiedemann, S. (2006). Conditional value-at-risk in stochastic programs with mixed-integer recourse. *Mathematical programming*, 105(2):365–386.

Sevi, H., Rilling, G., and Borgnat, P. (2018). Harmonic analysis on directed graphs and applications: from fourier analysis to wavelets. *arXiv preprint arXiv:1811.11636*.

Sha, F. and Saul, L. K. (2005). Analysis and extension of spectral methods for nonlinear dimensionality reduction. In *Proceedings of the 22nd international conference on Machine learning*, pages 784–791.

Shafipour, R., Khodabakhsh, A., Mateos, G., and Nikolova, E. (2018). Digraph fourier transform via spectral dispersion minimization. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6284–6288. IEEE.

Shalev-Shwartz, S. et al. (2012). Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194.

Shamsolmoali, P., Zareapoor, M., and Yang, J. (2019). Convolutional neural network in network (cnnin): hyperspectral image classification and dimensionality reduction. *IET Image Processing*, 13(2):246–253.

Shao, H., Lam, W. H., Sumalee, A., and Hazelton, M. L. (2015). Estimation of mean and covariance of stochastic multi-class od demands from classified traffic counts. *Transportation Research Part C: Emerging Technologies*, 59:92–110.

Shao, H., Lam, W. H., and Tam, M. L. (2006). A reliability-based stochastic traffic assignment model for network with multiple user classes under uncertainty in demand. *Networks and Spatial Economics*, 6:173–204.

Shapiro, A. (1989). Asymptotic properties of statistical estimators in stochastic programming. *The Annals of Statistics*, 17(2):841–858.

Shapiro, A., Dentcheva, D., and Ruszczynski, A. (2021). *Lectures on stochastic programming: modeling and theory*. SIAM.

Sharpnack, J., Singh, A., and Krishnamurthy, A. (2013). Detecting activations over graphs using spanning tree wavelet bases. In *Artificial intelligence and statistics*, pages 536–544. PMLR.

Sheffi, Y. (1985). *Urban transportation networks*, volume 6. Prentice-Hall, Englewood Cliffs, NJ.

Sherali, H. D., Sivanandan, R., and Hobeika, A. G. (1994). A linear programming approach for synthesizing origin-destination trip tables from link traffic volumes. *Transportation Research Part B: Methodological*, 28(3):213–233.

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98.

Singh, R., Chakraborty, A., and Manoj, B. (2016). Graph fourier transform based on directed laplacian. In *2016 International Conference on Signal Processing and Communications (SPCOM)*, pages 1–5. IEEE.

Smola, A. J. and Kondor, R. (2003). Kernels and regularization on graphs. In *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings*, pages 144–158. Springer.

Subbian, K. and Banerjee, A. (2013). Climate multi-model regression using spatial smoothing. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 324–332. SIAM.

Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V., and Gunopulos, D. (2006). Online outlier detection in sensor data using non-parametric models. In *Proceedings of the 32nd international conference on Very large data bases*, pages 187–198.

Sun, L. and Axhausen, K. W. (2016). Understanding urban mobility patterns with a probabilistic tensor factorization framework. *Transportation Research Part B: Methodological*, 91:511–524.

Suwansirikul, C., Friesz, T. L., and Tobin, R. L. (1987). Equilibrium decomposed optimization: a heuristic for the continuous equilibrium network design problem. *Transportation science*, 21(4):254–263.

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

Tan, H., Wu, Y., Shen, B., Jin, P. J., and Ran, B. (2016). Short-term traffic prediction based on dynamic tensor completion. *IEEE Transactions on Intelligent Transportation Systems*, 17(8):2123–2133.

Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323.

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

Transportation Networks for Research Core Team (2021). Transportation networks for research. *Transportation Network Test Problems. Available online: https://github.com/bstabler/TransportationNetworks (accessed on 10 September 2021).*

Unser, M., Aldroubi, A., and Eden, M. (1993). B-spline signal processing. i. theory. *IEEE transactions on signal processing*, 41(2):821–833.

Vardi, Y. (1996). Network tomography: Estimating source-destination traffic intensities from link data. *Journal of the American statistical association*, 91(433):365–377.

Vlahogianni, E. I. (2015). Computational intelligence and optimization for transportation big data: challenges and opportunities. In *Engineering and Applied Sciences Optimization*, pages 107–128. Springer.

Wang, D. Z., Liu, H., Szeto, W., and Chow, A. H. (2016a). Identification of critical combination of vulnerable links in transportation networks–a global optimisation approach. *Transportmetrica A Transport Science*, 12(4):346–365.

Wang, Y., Ma, X., Liu, Y., Gong, K., Henricakson, K. C., Xu, M., and Wang, Y. (2016b). A two-stage algorithm for origin-destination matrices estimation considering dynamic dispersion parameter for route choice. *PloS one*, 11(1):e0146850.

Wang, Y., Zhang, Y., Wang, L., Hu, Y., and Yin, B. (2021). Urban traffic pattern analysis and applications based on spatio-temporal non-negative matrix factorization. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):12752–12765.

Wang, Y.-X., Sharpnack, J., Smola, A., and Tibshirani, R. (2015). Trend filtering on graphs. In *Artificial Intelligence and Statistics*, pages 1042–1050. PMLR.

Wasserman, L. and Lafferty, J. (2007). Statistical analysis of semi-supervised regression. *Advances in Neural Information Processing Systems*, 20.

Watkins, D. S. (2007). *The matrix eigenvalue problem: GR and Krylov subspace methods*. SIAM.

Watson, J.-P., Wets, R. J., and Woodruff, D. L. (2010). Scalable heuristics for a class of chance-constrained stochastic programs. *INFORMS Journal on Computing*, 22(4):543–554.

Wen, C.-H. and Koppelman, F. S. (2001). The generalized nested logit model. *Transportation Research Part B: Methodological*, 35(7):627–641.

Wen, Z. and Yin, W. (2013). A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1):397–434.

Wiesemann, W., Kuhn, D., and Sim, M. (2014). Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

Wong, K., Wong, S. C., Tong, C., Lam, W., Lo, H. K., Yang, H., and Lo, H. (2005). Estimation of origin-destination matrices for a multimodal public transit network. *Journal of advanced transportation*, 39(2):139–168.

Wong, S. and Tong, C. (1998). Estimation of time-dependent origin–destination matrices for transit networks. *Transportation Research Part B: Methodological*, 32(1):35–48.

Xu, H. (2010). Uniform exponential convergence of sample average random functions under general sampling with applications in stochastic programming. *Journal of Mathematical Analysis and Applications*, 368(2):692–710.

Yang, C., Yan, F., and Xu, X. (2017). Daily metro origin-destination pattern recognition using dimensionality reduction and clustering methods. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 548–553. IEEE.

Yang, Y., Fan, Y., and Royset, J. O. (2019). Estimating probability distributions of travel demand on a congested network. *Transportation Research Part B: Methodological*, 122:265–286.

Yang, Y., Fan, Y., and Wets, R. J. (2018). Stochastic travel demand estimation: Improving network identifiability using multi-day observation sets. *Transportation Research Part B: Methodological*, 107:192–211.

Zehtabian, S. and Bastin, F. (2016). *Penalty parameter update strategies in progressive hedging algorithm*. Cirrelt Montreal, QC, Canada.

Zhang, Y., Duchi, J., and Wainwright, M. (2013). Divide and conquer kernel ridge regression. In *Conference on learning theory*, pages 592–617. PMLR.

Zhao, Y., Nasrullah, Z., and Li, Z. (2019). Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7.

Zheng, Z., Ahn, S., Chen, D., and Laval, J. (2011). Applications of wavelet transform for analysis of freeway traffic: Bottlenecks, transient traffic, and traffic oscillations. *Transportation Research Part B: Methodological*, 45(2):372–384.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI open*, 1:57–81.

Zhou, Y. and Tuzel, O. (2018). Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499.