The Pennsylvania State University
The Graduate School
Information Sciences and Technology

URBAN COMPUTING WITH MOBILITY DATA: A UNIFIED APPROACH

A Dissertation in
Information Sciences and Technology
by
Hongjian Wang

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2018

The dissertation of Hongjian Wang was reviewed and approved* by the following:

Jessie Li
Associate Professor of Information Sciences and Technology
Dissertation Advisor, Chair of Committee

C. Lee Giles
Professor of Information Sciences and Technology
Committee Member

Anna Squicciarini
Associate Professor of Information Sciences and Technology
Committee Member

Andrea Tapia
Associate Professor of Information Sciences and Technology
Graduate Program Chair

*Signatures are on file in the Graduate School.

Abstract

With the advent of information age, various types of data are collected in the context of urban spaces, including taxi pickups/drop-offs, tweets from users, air quality measure, noise complaints, POIs, and many more. It is crucial to use these data to understand the mobility-flow-incurred interactions in the city. This dissertation views the city as a spatial network of communities linked by mobility flow. Research questions that I try to answer are: 1) how to understand nodes using heterogeneous links; 2) how to automatically partition continuous space to construct discrete nodes; 3) how to model spatial non-stationary property within the spatial network.

This dissertation aims at modeling the complicated interactions of regions in the urban space. Traditionally, due to lack of flow data, interaction is defined only by spatial distance. Recently, the availability of movement data enables us to study the interactions incurred by various social flow data (e.g., taxi flows, commuting flows). While various approaches are proposed, they still have the following drawbacks. 1) The definition of interactions from different data sources is ad-hoc. 2) Rely on a predefined region boundary which may not be the appropriate. 3) Assume uniform correlation across space.

In this dissertation I proposes to develop a unified framework to model the mobility-flow-incurred interactions in the urban context. We start with a preliminary study on improving the Chicago community level crime prediction with POI and taxi flow. Second, We propose a heterogeneous graph embedding method to incorporate various links and define region similarity. Third, we propose a method to automatically partition the city into regions, in order to preserve the consistency of correlations among different features. Finally, we tackle the spatial non-stationary property with local models instead of a global model.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to express the deepest appreciation to my doctoral committee Dr. Zhenhui Li, Dr. C. Lee Giles, Dr. Anna Squicciarini, Dr. Daniel Kifer, and my Ph.D. program chair Dr. Andrea Tapia.

I would like to express my special gratitude to my adviser Dr. Zhenhui Li, for her professional mentorship and collaboration, for her continued support and understanding, and for acting as the role model that we look up to. My gratitude also extends to Dr. Corina Graif from Department of Sociology and Criminology, Dr. Wang-Chien Lee and Dr. Daniel Kifer from Department of Computer Science and Technology. Their support during our collaboration gives me that extra edge to deal with my research challenges. I am also thankful to my labmates and other Ph.D. students from my cohort for their various inputs on the problem that I am working on.

The research work in this dissertation would not have been possible without funding support from the National Science Foundation. The work was supported in part by NSF award #1544455, #1618448, #1054389, #1652525, #1639150, and funding from NICHD R24-HD044943.

Lastly I would also like to thank my parents and friends who supported me during the twists and turns in the past five years. Especially, I want to thank Haining for making me strong.

Chapter 1
Introduction – Urban Computing

1.1 Urban Computing

Urbanization's rapid progress has led to many mega cites, where people's lives have been enriched with advanced technology. Meanwhile new issues, such as air pollution and traffic congestion, pose greater challenges to us. To address these challenges, *urban computing* [1] emerges as a means to unlock the power of big data from modern cites, to tackle the challenges faced by modern cities, and to build more intelligent cities.

Nowadays, sensing technologies and large-scale computing infrastructure have produced a variety of *big data* in urban spaces. In Figure 4.1, we present an example of various data available in the City of New York. The big data have three major properties.

1. Data variety. All kinds of data are collected in different formats. For example, the crime data is reported at incident level with detailed time and location. Meanwhile, the weather data and air quality data are available at the whole city scale.

2. Data volume. The quantity of urban data could be huge. For example, there are over 450,000 taxi trips are made in New York City every day. There are over 380,000 venues in New York City, where people can visit and write reviews online.

3. Data velocity. The speed of new data being generated is fast. On average, there are 621 millions of tweets are generated every data.

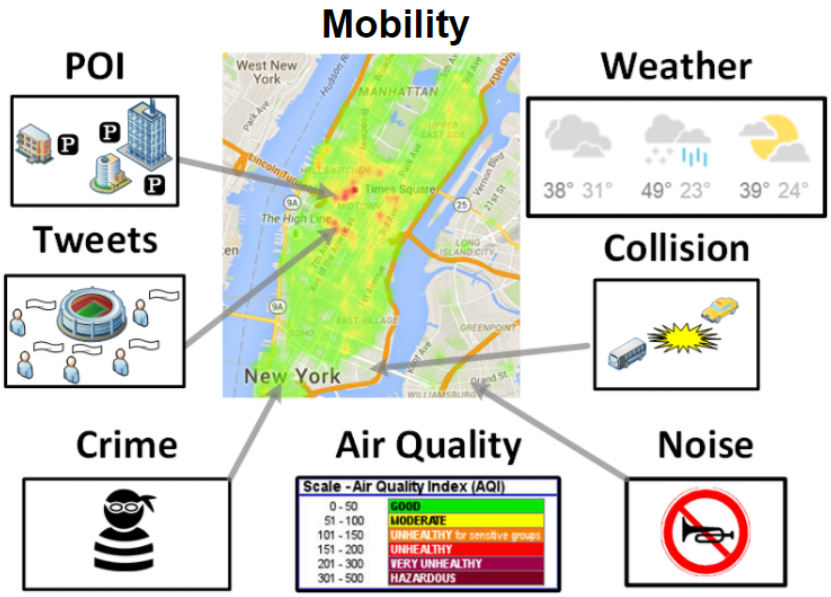


Figure 1.1. Various big data collected in urban space.

All those urban data provides amazing opportunities for us to study urban problems and build intelligent urban applications. Among all these data, the urban mobility data is especially interesting because the mobility flow connect a pair of locations.

1.2 Model Mobility-Flow-Incurred Interactions

In the urban space, the movement of human population connects two disjoint regions, and brings influence from one to the other. My research focuses on modeling the complicated interactions in the urban space with human mobility data.

In recent years, there are some large datasets of urban taxi made public [2] under the Freedom of information request law. The taxi dataset contains the time and location of pick-up and drop-off for a trip. By aggregating the taxi data, we are able to get the mobility flow among different regions. Mobility flow data (e.g., commuting flow, taxi trajectories) are sensitive resources that urban planners can use to address city issues.

Consider the following two examples. These two examples show that it is not sufficient to only consider spatial interaction, and the interactions incurred by mobility flow is equally important.

Example 1. *Policy makers are deciding where to construct a shelter for families that are victims of violence. They understand the value of locating the shelter geographically far from violent neighborhoods. One possible choice is to locate the shelter in a neighborhood that is 10 miles from the violent neighborhoods where vulnerable families previously lived. However, a deeper analysis may reveal that the new neighborhood, though geographically removed from the old neighborhood, may still have strong social flows (connections caused by commutes, family visits) with the old neighborhood. Emerging research suggests that a great deal of crime happens in areas that are socially connected to offenders' neighborhoods. This suggests that shelters may benefit from being located in a neighborhood that is also socially isolated from violence (e.g., with weak communication and commuting interactions with the violent neighborhoods that shelter residents fled from) while socially connected to jobs, services, and resources.*

Example 2. *In order to provide better living environment to crowded city, city planner decide to expand city with a new satellite town. However, people are not willingly to move, if there is not enough incentive. To create opportunity and attract people, some candidates to relocate are big factory, shopping mall, and government offices. The question is which facility should be relocated, so that population pressure in whole city is relieved. Correspondingly, how many new residential building, supermarket, and parks should be built into the new city? People have to go the new city to work. Some of them may move to live in the new city, and the rest may choose to commute everyday.*

1.3 Research Problems

In this dissertation, we model the city as a spatial network of regions, where each region is one node and there is a link between a pair of nodes. Links measure the interactions between a pair of regions. For example, the spatial link defines the spatial adjacency among regions. There are other types of links as well. Accounting for these different types of region **interactions** can improve the a set of urban prediction tasks. I am specifically interested in the mobility flow data because they

act as a type of "hyperlink" and connect regions that are spatially far away. In order to model the city as a spatial network, we have to answer the following three research questions.

- How to better understand nodes using links? In an urban prediction task, we usually estimate an unobserved property of a focal region from the observations of similar regions. The challenge lies in an appropriate definition of "similar regions". In the literature, spatial similarity is widely used. However, we argue that mobility flow also plays an important role in defining region similarity. For example, the crime rate in a residential neighborhood could be impacted by spatially non-adjacent but flow-connected neighborhoods.
- How to define appropriate nodes in the spatial network? The administrative boundaries are widely used to define discrete regions. However, there are also a lot of concerns with the predefined boundaries. What if we do not have the boundaries available? What if the current boundary is outdated? Is the predefined boundary suitable for our prediction task?
- What is the appropriate model for the correlations among various data in our spatial network? We argue that one global model cannot necessarily fit all scenarios. For example, nationwide we may observe that the house price is negatively correlated with house density. However, such observation is not universally true because we can easily find counterexample. For example, the house price in Manhattan island is high regardless of the high house density.

1.4 Interactions within Networks in the Literature

1.4.1 Spatial Interaction Model

Spatial interaction is a broad term encompassing any movement over space that results from a human process [3, 4]. It measures the flow between an origin and a destination, given distance and nodal properties on the origin and destination. There are several classic models to measure the spatial interactions. For example, **gravity model** [5] uses a similar formulation than Newton's law of gravity to model the interaction between two regions. **Spatial autoregressive model** [6] is another widely used inference model that infer a property in a region with

nearby region's information. There are a lot of applications based on the spatial interactions, such as international trade [7–9], population migration [10–12], traffic flow [13–15], telecommunication flow [16–18], crime estimation [19–21], knowledge spillover [22, 23], and many more.

Notice that the region interaction in this thesis proposal is different from the traditional **spatial interaction** study, which assumes the interaction of two regions is reversely correlated with their geographical distance. However, space is a biased sample on human mobility, and should not be the only measure to model human movement. For example, during the weekdays people usually follow a regular home-office commuting pattern. The workplace region might be far from home region, but their connection is strong due to the human movement. As there are more and more mobility data available, we propose to employ the flow data in the interaction model as well.

1.4.2 Exploring Mobility Flow

In recent years, the availability of flow data enables research progress on exploring the role of various flows. New observations are made on the mobility data. For example, there are studies to explain epidemic spread with air travel [24, 25] and population migration survey [26]. Zheng et al. [27] identify the underlying road network problem by detecting anomalous pair of flow. Yuan et al. [28] discover the function of regions by learning a topic model on the flow matrix and clustering the flow of region pairs into different topic. Berlingerio et al. [29] optimizes public transport route by looking at the region origin/destination flow.

The problem in this proposal is different from the works above in the sense that different approaches are taken. Those works in the literature study the mobility flow by itself. Namely, the literature focuses on observing the properties of the mobility flow by various data mining technique, such as clustering [29], topic modeling [28], outlier detection [27] etc. The approach in the literature uses the observed properties of flow to manually explain a phenomenon. This thesis proposal takes a different approach, that explicitly models the statistic dependency between social flow and region properties in an inference problem setting. The approach in this thesis allows us to answer those three research questions raised in Section 1.3, which the literature methods cannot answer.

1.4.3 Interaction Model in Social Networks

In social network analysis, there is a line of work that focuses on infer an unobserved nodal feature from neighboring nodes. Examples are infer user home location [30,31], user age [32], and more categorical features [33]. Earlier works mainly focus on one type of features, and employ the network structure to make inference. To solve the application problem of user profiling, some techniques are borrowed from community detection [34], information diffusion [35], collaborative filtering [36], etc. The inferred features are assumed to be similar within a cluster. Therefore, community detection will cluster nodes with unobserved property together with nodes with observed property, and thus we can conduct inference. Another angle is to study how information is propagated through network. Methods under the information diffusion category is explanatory [37, 38].

With more data types available, there is one line of works solving the nodal feature inference problem with **composite social network** [39–41]. Composite social network is a graph with multiple types of edges. For example in a social network, two users could be connected by following, how many re-tweets, how many messages are exchanged, etc.

Our work is different from those work in social network literature in a way that we do not use observed edge strength as the interactions. All those paper in literature assumes that if two nodes are connected by a edge with heavy weight, then they are very likely to have strong interactions. This is not necessarily true. Take the crime inference as example. Suppose two communities are connected by heavy traffic flow, and both community are crime free. In this case, we cannot say two regions have strong crime interactions.

1.5 Challenges

There are mainly three challenges to address.

Given multiple types of social flow, the interaction is difficult to define. Given only one type of social flow, there are too many possibilities in constructing interactions. First, the flow matrix can take various form. For example, we can chose normalize the flow matrix or not. When normalizing the flow matrix, we can chose whether normalize by in-flow or out-flow. Second, there are many nodal

properties that could interact with their neighbors, such as various demographics features. Third, different kinds of function can be used to define interactions. Take the product of flow and nodal properties is the most straightforward choice. However, sometimes it also makes sense to further apply distance exponential decay on previous product. Furthermore, it is possible we have multiple types of social flow (e.g. taxi flow, commuter transit). For one pair of regions, should we build separate interaction over different social flows, or sum over all flows to get one interaction? When we take the sum, should we weight different social flow differently, and how?

Identify discrete node from continuous space is challenging. In the literature, the boundary of regions are usually pre-defined administrative regions, which has two major issues. First, this boundary definition is not consistent in terms of the property of interest. For example, it is very likely a community is formed on the boundary of two administrative regions. Also, the current partition does not evolve over time. In real life, we know there must be some dynamic change of communities. Second, partition continuous space is not trivial. There is a misalignment problem, which refers to the problem that different data are not collected in the same scale. For example, the crime record is at point level, while the demographics is at block level.

The interaction is non-stationary over the space. Most models in existing work are global model, which assumes the statistic interaction does not vary over space. However, some urban data have spatial non-stationary property. Therefore, using global estimates of relationships can present misleading interpretations of local relationships.

To address these challenges, I have three : (1) use graph embedding method to incorporate heterogeneous region interaction data; (2) use geographically weighed graphical model to account for the spatial non-stationary property; (3) automatically learn task-specific community area partitions.

1.6 Organization

The rest of my dissertation is organized as shown in Figure 1.2. In Chapter 2 I give a preliminary study to verify the idea that using mobility flow as hyper-links to

better model region similarity. We use crime inference as an example, in which we use an enhanced spatial autoregressive model to predict crime count of a community using its neighbors.

To address the aforementioned three challenges, I have correspondingly three major components in my dissertation. I propose an embedding method to incorporate all kinds of region interactions in Chapter 3. Next, I study the task-specific region partition problem in Chapter 4 to tackle the spatial continuity. In Chapter 5, we first identify the spatial variations in mined relations. We further employ a geographically weighted regression model to solve the problem. Finally, I conclude in Chapter 6 with a discussion of potential future topics.

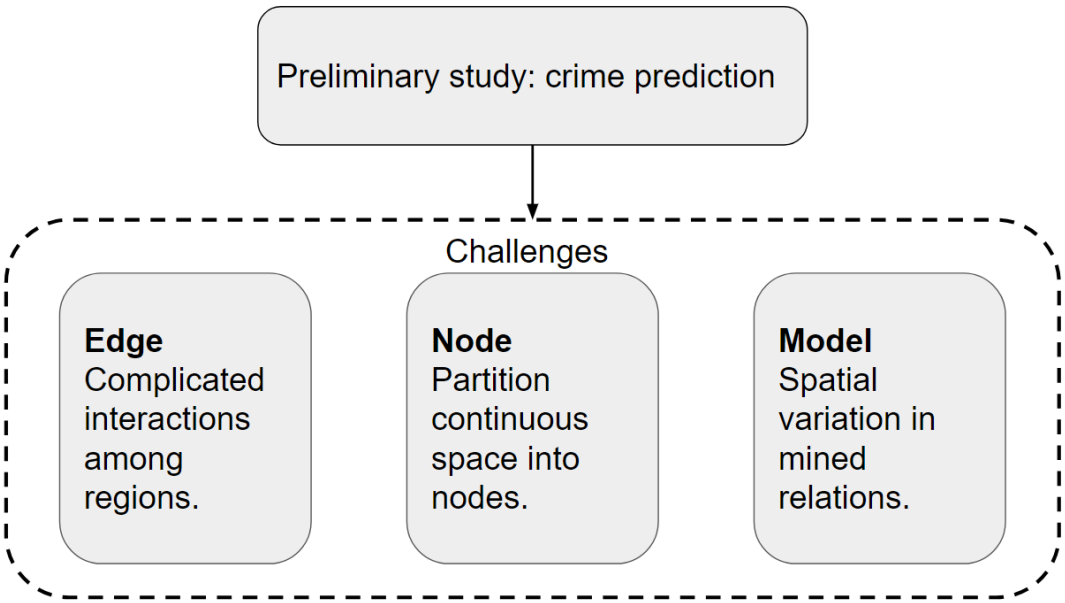


Figure 1.2. Organization of this dissertation.

Chapter 2
Crime Rate Inference with Big Data

In this chapter we elaborate the crime rate inference problem as a preliminary study. The main takeaway is that the heterogeneous urban data, especially the mobility data, are helpful for understanding urban properties.

2.1 Introduction

Understanding how to control crime is important because exposures to violence and crime have been unusually high in the U.S. for several decades and, while declining, they remain high [42, 43]. Over half a million children and youth aged 10-24 years were treated in 2012 in emergency departments for nonfatal physical assault injuries related to gun shots, cuts and stabbings, among others [44]. Understanding the neighborhood context of crime is particularly important because victimization and other forms of crime exposures have many severe consequences. Beyond the high medical bills and violent death, consequences include behavioral and mental health problems, aggression, substance abuse, post-traumatic stress disorder, and suicide, lower academic achievement, and engaging in further violence [45].

In this paper, we study the problem of crime rate inference of communities. We select Chicago as the target of study for the following reason. Chicago has more homicides and non-negligent manslaughter rates (15.2) per 100,000 residents than New York (4.0) and Los Angeles (6.5) according to the FBI crime statistics for 2013 and has experienced no decline in the past decade compared to the other two large cities, which have been on a slow declining slope [46].

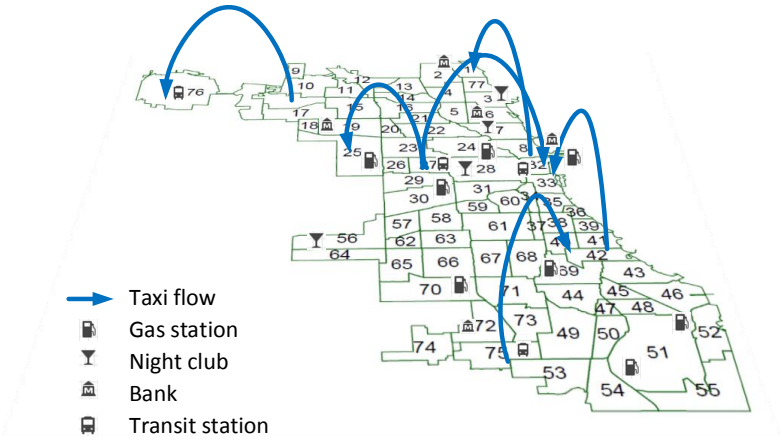


Figure 2.1. An illustration of various types of features we used in Chicago. The POI distribution across community areas reflects profiles of the region functionality. The taxi flow connects non-adjacent regions and act as "hyperlinks" on the space.

Traditionally, researchers have used demographic information (e.g., population poverty level, socioeconomic disadvantage, racial composition of population) to estimate the crime rate in a community [47]. However, such demographic information only contains partial information about the neighborhoods and does not dynamically reflect the changes in the community (e.g., official counts are collected by the U.S. Census Bureau every 10 years). Using only demographic information will result in a relative error of at least 30% for crime rate estimation in Chicago (refer to experiment section in the paper). Existing studies also use the geographical influence [48] to estimate the crime rate, i.e., the crime in the nearby communities can be propagated to the focal community. But this geographical influence is of little help in improving the crime inference on top of demographic feature, with at most 0.4% relative improvement in our experiments. This is probably because the nearby communities also share similar demographics, which limits the additional benefit of geographical influence.

Recently, big data reflecting city dynamics have become widely available [49], e.g., traffic flow, human mobility, social media, and crowd-generated Points-Of-Interest (POI). As shown in Figure 2.1, such newer types of big data could provide us new insights to understand some traditional socioeconomic urban problems, such as the crime rate inference problem we focus on in this paper. In particular, we propose to study two newer types of urban data: POI and taxi flow.

POI data. POI data provide venue information such as GPS coordinates, category, popularity, and reviews. These POIs mostly belong to categories such as food, shop, transit, education, etc. Recent studies have shown that using such categorical information of POIs are useful to profile neighborhood functions [50]. Such neighborhood functions could further help us predict crime rate (e.g., communities with less education or entertainment facilities may have a higher rate of crime). Our experiments show that incorporating POI features significantly improve the crime rate inference. Adding POI features in addition to demographics features reduces the relative error by at least 5% in our experiments. This demonstrates that POI data provide additional information about the communities that is not covered by the demographics.

Taxi flow data. A huge amount of taxi flow data reflect how people commute in the city. In previous studies, when using geographical influence [48], people assume that a community is affected by the spatially nearby communities. However, communities are not only affected by spatially-close communities. Even if two communities are distant in geographical space, they could have a strong correlation if many people frequently travel between these two communities [51]. We hypothesize that taxi flows may be considered as "hyperlinks" in the city that connect the locations and we use such data to estimate crime rates. Taxis may be preferred to public transportation by offenders traveling to a crime location as they offer more privacy and more flexible pick-up and drop-off points. Even if taxis do not constitute the main transportation mode in committing crime, taxi flows may be a proxy for broader patterns of population routine activity and mobility, commuting flows, and other forms of social and economic exchanges between two communities over space. Such exchanges may increase the number of potential targets and opportunities for crime [52, 53] or contribute to inter-community diffusion of information about successful local strategies to control or prevent crime (e.g., successful features of neighborhood watch programs). Our experiments show very promising results – adding taxi flow data on top of all other features can further decrease the error by 5%.

We conduct extensive experiments including a systematic comparison between linear regression and negative binomial models, tests of different combinations of features, detailed discussions of how to construct features, analysis of the relative importance of features, and theoretical interpretations of the results from a social

scientist (a co-author in the paper). The experiments are conducted on the crime data over multiple years. We demonstrate that using the big urban data shows significant improvements.

In summary, the contribution of this paper are: 1) We study an old but very important crime inference problem by utilizing new urban data: POIs and taxi flows. 2) We find that utilizing these new types of big urban data significantly improves the crime rate inference. 3) We conduct systematic experiments to compare different results and feature combinations. The significantly better performance could serve as a new baseline for future crime inference problems.

The rest of this chapter is organized as follows. The crime inference problem is formulated in Section 2.2. We discuss the inference model in Section 2.3 and feature extraction procedure in Section 2.4. The Section 2.5 presents the quantitative evaluation results on real data. We review the related work in Section 2.6. Finally, we conclude this chapter in Section 2.7.

2.2 Overview

The crime data collected in Chicago has detailed information about the time and location (i.e., latitude and longitude) of crime and the types of crime. In our problem, the term crime count refers to number of crime incidents in a region (i.e., community area) in a year. The *community area* is used as our geographical unit of study, since it is well-defined, historically recognized and stable over time [54]. In total, there are 77 community areas in Chicago. Crime rate is the crime count normalized by the population in a region. We use vector $\vec{y} = [y_1, y_2, \ldots, y_n]$ to denote the crime rates in regions. The crime rate inference problem is to estimate the crime rate in one region using the crime rate of other regions in the same year by considering the features of regions and correlations between regions.

The crime data of Chicago are obtained from the City of Chicago data portal [55]. Chicago is the city with most complete crime data that are made public online. The crime dataset contains the incident date, location (strict name and GPS coordinates), and primary type from year 2001 to 2015. In total there are 5,856,414 recorded crime incidents over 15 years, which is an average 390,417 crimes incidents per year. We visualize the crime normalized by population in Figure 2.2, from which we can see that the downtown area has the highest crime rate.

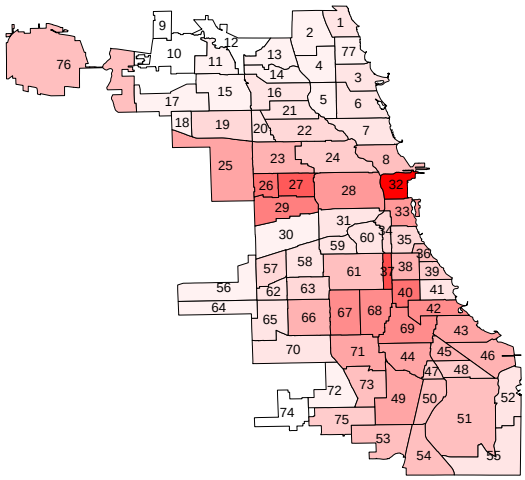


Figure 2.2. Crime rate of Chicago by community areas. The community area #32 is Chicago downtown, which has the highest crime rate.

In this paper we study the crime rate inference problem. More specifically, we estimate the crime rate of some regions given the information of all the other regions. Without loss of generality, we assume there is one community area t with crime rate y_t missing, and we use the crime rate of all the other regions $\{y_i\}\backslash y_t$ to infer this missing value. Our problem is mathematically formalized as follows

$$\hat{y_t} = f(\{y_i\}\backslash y_t, X), \tag{2.1}$$

where X refers to observed extra information of all those community areas.

We consider two types of features X for inference:

- Nodal feature. Nodal features describe the characteristics of the focal region. Such features include demographic information and Point-of-Interest (POI) distribution. Demographics are frequently used in literature, but POI is a newer type of big data, which we find significantly improve the crime inference accuracy.

- Edge feature: (1) Geographical influence. Geographical influence considers the crime rate of the nearby locations. This feature has been extensively used in literature as well. To estimate the focal region, the crime rate of nearby regions are weighted according to spatial distances. (2) Hyperlink by taxi flow. Locations are connected through the frequent trips made by humans, which can

be considered as the hyperlinks in space. This type of feature has never been studied in literature. We propose to use taxi trips to construct the social flow. Our hypothesis is that two regions that are more strongly connected through social flow will influence each other's crime rate.

In the following sections, we first discuss the inference models based on these three types of features in Section 5.1 and then discuss how to construct these features using the real-world data in Section **??**.

2.3 Inference Model

2.3.1 Linear Regression

The most straightforward prediction model is the linear regression. This model assumes the error terms follow a Gaussian distribution $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

Equation 5.1 gives the linear regression formulation of our problem.

$$\vec{y} = \vec{\alpha}^T \vec{x} + \beta^f W^f \vec{y} + \beta^g W^g \vec{y} + \vec{\epsilon}, \tag{2.2}$$

where $\vec{x}$ represents the nodal features, including demographics and POI distribution, W^f is the flow matrix of taxi flow, and W^g is the spatial matrix representing the geographical adjacency. On the right-hand side, ϵ is the only stochastic variable, and all other terms are fixed observation values. Therefore, we incorporate all the fixed observations into one term X, and we get the standard regression problem

$$E(y) = Xw + \epsilon.$$

2.3.2 Negative Binomial Regression

In our problem, we aim to infer the crime rate, which is guaranteed to be a non-negative integer. However, linear regression does not ensure this property. *Poisson regression* is another form of regression, more appropriate for count data than linear regression [56] [57]. With shortened notation X, the Poisson regression model has the exponential function as link function

$$E(y) = e^{Xw}. \tag{2.3}$$

This comes from the assumption that y follows Poisson distribution with mean λ. Additionally, the mean λ is determined by observed independent variables X, with the link function $\lambda = e^{Xw}$. Adding all together, the joint probability of y is

$$P(y|w) = \frac{e^{-e^{Xw}}(e^{Xw})^y}{y!}. \tag{2.4}$$

However, Poisson regression enforces the mean and variance of dependent variable y to be equal. This restriction leads to the "over-dispersion" issue for some real problems, that is the presence of larger variability in data set than the statistical model expected. To address this, we use the Poisson-Gamma mixture model, which is also known as *negative binomial regression*. Negative binomial regression has been used in similar work [58].

Given that the crime rate y follows Poisson distribution with mean λ, in order to allow for larger variance, λ itself is a random variable having a Gamma distribution with shape $k = r$ and scale $\theta = \frac{1-p}{p}$. The probability function of y becomes

$$\begin{aligned} P(y|r,p) &= \int_0^\infty P_{Poisson}(y|\lambda) \cdot P_{Gamma}(\lambda|r,p)d\lambda \\ &= \frac{\Gamma(r+y)}{y!\Gamma(r)}p^k(1-p)^y \end{aligned} \tag{2.5}$$

This is exactly the probability density function of negative binomial distribution.

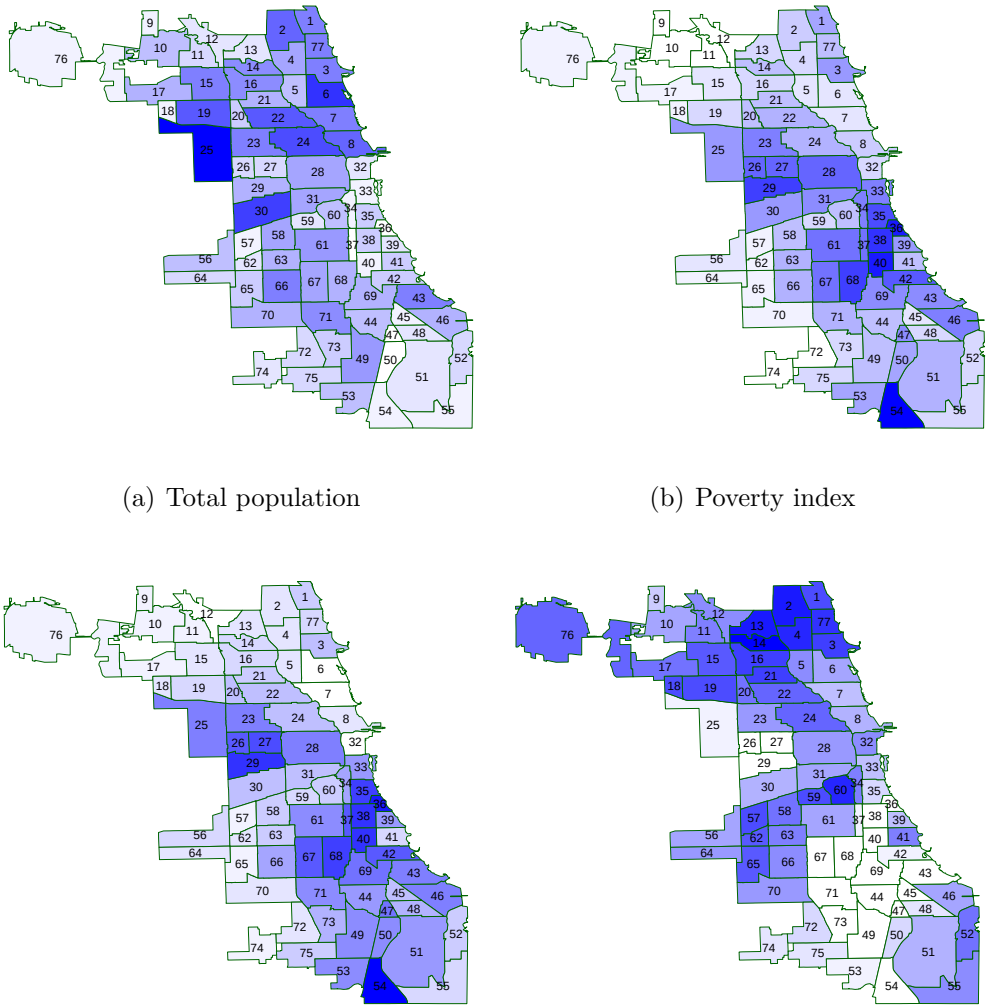
In negative binomial regression, the link function is

$$E(y) = e^{Xw+\epsilon}. \tag{2.6}$$

The error term e^ϵ is the mixture prior, and we assume it follows Gamma distribution with shape parameter $k = \frac{1}{\theta}$, so that it has mean $E(e^\epsilon) = k\theta = 1$ and variance $Var(e^\epsilon) = k\theta^2 = \theta$. This setting ensures the $E(y) = e^{Xw} \cdot e^\epsilon = e^{Xw}$.

2.4 Feature Extraction

In this section, we will discuss the details of features used in our method. The two types of new features we use are extracted from Point-Of-Interest data and taxi flow data. Below we describe the datasets used to construct features and the characteristics of these features.



(a) Total population

(b) Poverty index

(c) Disadvantage index

(d) Ethnic diversity

Figure 2.3. (a)-(d) Demographics in Chicago by community areas. Darker colors indicate higher values. Each demographic feature is normalized into $[0, 1]$.

2.4.1 Nodal Feature: Demographics

Socioeconomic and demographic features of neighborhoods have been widely used to predict crime [59–62]. Previous studies have shown that crime rate correlates with certain demographics. For example, [47, 63] suggests that population diversity leads to less crime in certain neighborhoods. In our study, we include demographic information from the US Census Bureau's Decennial Census [64]. Using 2010 census

information would overlap with the time in which crime is measured. Instead, we use year 2000 demographic data because we are interested in predictors that precede temporally the period in which crime rates are evaluated. The demographics include the following features:

total population, population density, poverty, disadvantage index, residential stability, ethnic diversity, race distribution.

The poverty index measures the proportion of community area residents with income below the poverty level. The disadvantage index is a composite scale based on prior work [65], a function of poverty, unemployment rate, proportions of families with public assistance income, and proportion of female headed households. The residential stability measures home ownership and proportion of residents who lived in the neighborhood for more than one year. Racial and ethnic diversity is an index of heterogeneity [47] based on six population groups, including: Hispanics, non-Hispanic Blacks, Whites, Asians, Pacific Islanders and others.

Figure 2.3 visualizes the crime rate and demographics features in Chicago by community areas. Comparing with Figure 2.2, it is clear that the crime rate and poverty index and disadvantage index are consistent, the ethnic diversity shows an inverse correlation, and the total population has little correlation with crime.

Table 2.1 shows the Pearson correlation coefficient between various demographics features and the crime rate at community area level. The corresponding p-value is also calculated and shown in the table to indicate the significance of the correlation coefficient. There are in total 77 community areas in Chicago. Table 2.1 shows such correlation with several most correlated features. We can see that the poverty index and disadvantage index positively and strongly correlate with crime, while the ethnic diversity negatively correlates with crime. Other features such as total population, population density, and residential stability have weaker correlations. One counter-intuitive observation is that the total population has a weak and negative correlation with crime. The reason is that we use crime rate in each community area, which is already normalized by the population, and therefore the total population and population density have less impact.

Table 2.1. Pearson correlation between demographic features and crime rate (***** indicates significant correlations with p-value less than 5%).

Feature	Correlation	p-value
Total Population	-0.1269	0.2716
Population Density	-0.1972	0.0855
Poverty Index	**0.5573***	1.403e-07
Disadvantage Index	**0.5959***	1.082e-08
Residential Stability	-0.0453	0.6965
Ethnic Diversity	**-0.5545***	1.678e-07
Percentage of Black	**0.6696***	2.779e-11
Percentage of Hispanic	**-0.3820***	0.0006

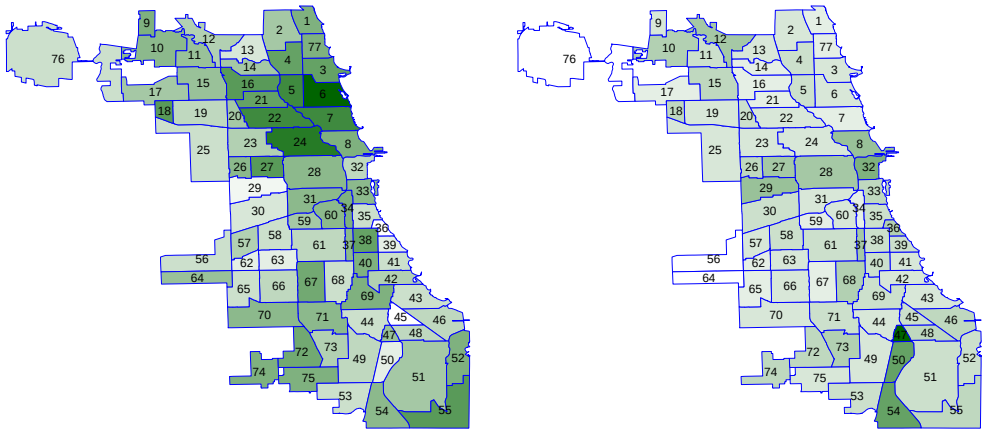
2.4.2 Nodal Feature: Point-of-Interest (POI)

While demographics are traditional census data, POI is a type of modern data that provide fine-grained information about locations. We collect POI from FourSquare [66]. POI data from FourSquare provide the venue information including venue name, category, number of check-ins, and number of unique visitors. We mainly use the major category information because categories can characterize the neighborhood functions. There are 10 major categories defined by FourSquare:

food, residence, travel, arts & entertainment, outdoors & recreation, college & education, nightlife, professional, shops, and event.

In total, we have crawled 112,000 POIs from FourSquare for Chicago. Most of these POIs are in the downtown area of Chicago. For the purpose of visualization, we normalize the POIs count per category by the total POI count in a neighborhood and plot two selected categories, i.e. nightlife and professional, in Figure 2.4. The darker colored neighborhoods in Figure 2.4 are the ones with a higher proportion of residence POIs.

In Table 2.2 we show the Pearson correlation between POI category and crime rate. The category "professional" is most significantly correlated with the crime rate. Under the professional POI category, there are some venues with a large population concentration, such as transportation center, convention center, community center, and coworking space. In those venues, the population volume is high and residential stability is low, therefore the professional POI counts positively correlates with crime rate. One counter-intuitive observation is that "nightlife" category is not positively correlated with crime (-0.1553). This can be seen in Figure 2.4(a). The majority of nightlife venues in Chicago are located in the northern area, while most



(a) Nightlife (b) Professional

Figure 2.4. POI ratio per neighborhood. The saturation of color is proportional to the ratio value. The "professional" category distribution is more consistent with the crime distribution, and therefore it is the most correlated with crime. Meanwhile, the "nightlife" category is negatively correlated with Chicago crime. The POI ratios are independently normalized for different POI categories.

crime incidents occur in the downtown area.

2.4.3 Edge: Geographical Influence

Together with the US census demographics data, we also collected the boundary shape files of Chicago, which are used to calculate the geographical influence feature. Previous studies have also shown that the crime rate at one location is highly correlated with nearby locations [67, 68]. Such geographical influence is also frequently used in the literature [69, 70]. It is calculated as:

$$\vec{F^g} = W^g \cdot \vec{Y}, \tag{2.7}$$

where W^g is the spatial weight matrix. If region i and j are not geospatially adjacent, $w^g_{ij} = 0$; otherwise, $w^g_{ij} \propto distance(i,j)^{-1}$.

In Figure 2.5, we plot crime rate with respect to geographical influence calculated in Eq. 2.7. We observe an obvious positive correlation, which means if nearby neighborhoods have a high crime rate, the focal neighborhood is more likely to have a high crime rate. We also do observe a few outliers in Figure 2.5. These

Table 2.2. Pearson correlation between POI category and crime rate (* indicates significant correlations with p-value less than 5%).

POI category	Correlation	p-value
Food	-0.1543	0.1803
Residence	-0.0610	0.5984
Travel	-0.0017	0.9883
Arts & Entertainment	-0.0049	0.9661
Outdoors & Recreation	0.0668	0.5637
College & Education	-0.0078	0.9473
Nightlife	-0.1553	0.1775
Professional	**0.3221***	0.0043
Shops	-0.1676	0.1450
Event	0.2196	0.0549

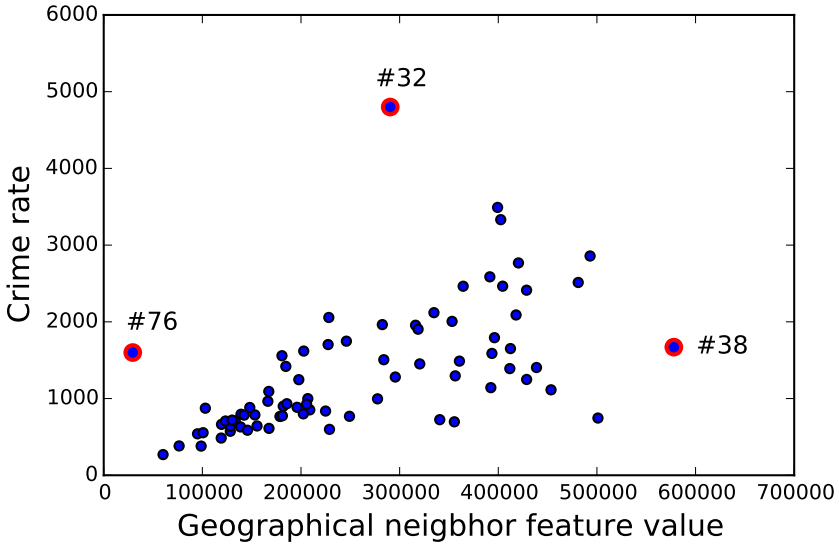


Figure 2.5. The correlation between geographical influence feature and crime rate. In the plot we marked out three outliers and their corresponding community area ID.

neighborhoods show different crime rate in their nearby neighborhoods compared to their own. For example, as we can also see in Figure 2.2, community area #38 locates in an area where the the neighbors have high crime rates but its crime rate is relatively low; in contrast, neighborhood #32 has a high crime rate even though its neighbors have relatively low crime. The community area #76 home of the O'Hare International Airport is far from most of other community areas, however

its own crime rate is relative high.

2.4.4 Edge: Hyperlinks by Taxi Flow

In our Chicago taxi dataset, there are 1,048,576 taxi trips in total from October to December in 2013. For each trip, the following information are available: pickup/dropoff time, pickup/dropoff location, operation time, and total amount paid. We requested the taxi trip records from Chicago under the Illinois Freedom of Information Act. Figure 2.6 shows a visualization of the major flows at community level.

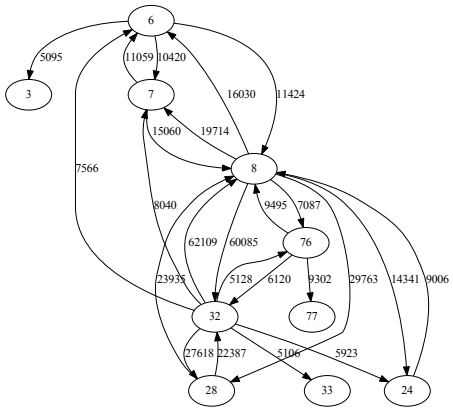


Figure 2.6. Major taxi flows between neighborhoods. The label on the edge shows the count of taxi trips commuting between two community areas from October to December months in 2013. We set a threshold (more than 5,000 trips) on the flow and only plot high volume flows. The label on a node is the ID of its corresponding community area. We can see that there are several hub community areas, such as #6, #8, #32, which are all in the downtown areas.

One of our hypotheses is that the social interaction among two community areas propagates crime from one region to another. The Chicago taxi data captures the social interactions among various community areas. To calculate this, we first map all taxi trips to community areas to get the taxi flow w_{ij} $\forall i, j \in \{1, 2, \cdots n\}$. Then the taxi flow lag is constructed by the product of social flow and the crime rate of neighboring regions as follows

$$\vec{F^t} = W^t \cdot \vec{Y}. \tag{2.8}$$

The taxi flow W^t is a matrix with entry w_{ij} denoting the taxi flow from i to j. Note that $\forall i$, $w^s_{ii} = 0$ in matrix W^t, because we have to exclude the crime in the focal area from its own predictor. The semantic of this taxi flow feature is how much crime in the focal area is contributed by its neighboring areas through social interaction.

The correlation between taxi flow and crime rate is shown in Figure 2.7. From the scatter plot, we can see that overall the crime rate is positively correlated with the taxi flow. There are two outliers clearly shown in Figure 2.7. The community area #32 is the downtown Loop, which has the highest crime rate and is hard to predict by taxi flow. Another anomalous community area (#47) has relatively low crime rate by itself. However, this area has a lot of in flows from high-crime communities.

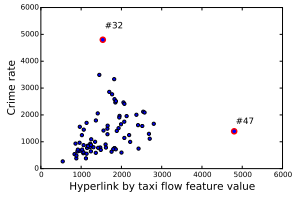


Figure 2.7. Correlation between taxi flow feature and crime rate. In the plot, we marked out two outliers and their corresponding community area ID.

2.5 Experiments

2.5.1 Settings

We adopt leave-one-out evaluation to estimate the crime rate of one geographic region given all the information of all the other regions. When we construct the spatial/social lag variable for the training data, the effect of testing region is completely removed. For example, if region y_t is the testing region, the remaining $\{y_i\}\backslash y_t$ become the training set. For any y_j in the training set, its geographical influence feature and taxi flow feature are constructed from $\{y_i\}\backslash\{y_t, y_j\}$.

In the evaluation, we estimate the crime rate for testing community areas. The accuracy of estimation is evaluated by mean absolute error (MAE) and mean relative error (MRE).

$$MAE = \frac{\sum_i^n |y_i - \hat{y}_i|}{n} \tag{2.9}$$

$$MRE = \frac{\sum_i^n |y_i - \hat{y}_i|}{\sum_i^n y_i} \tag{2.10}$$

2.5.2 Performance Study

We evaluate the estimation accuracy under various feature combinations. The leave-one-out evaluation results are shown in Table 2.8. We run both the linear regression and the negative binomial regression on five consecutive years, 2010 – 2014. Both MAE and MRE are shown in the table. We have four types of features: demographics, POI, geographical influence and taxi flow. We test the various settings of feature combinations.

2.5.2.1 Negative Binomial Regression vs. Linear Regression

In Table 2.8, we can see that in different years and under most settings, the negative binomial regression significantly outperforms the linear regression (with only a few exceptions when using only demographic feature). When using all the features, NB is significantly better than LR with at least 6% improvement in relative error. One reason is that negative binomial regression is a count prediction model, which guarantees the prediction variable is non-negative . Another reason is that it is difficult to get very precise estimates of crime rate, and the negative binomial regression allows a large variance in the estimated crime rate. Therefore negative binomial is more appropriate for crime rate estimation than linear regression.

In the following discussions, we only refer to the performance of the negative binomial regression.

2.5.2.2 POI Feature

Adding POI features always improves the accuracy (see NB for column 2 vs. column1, column 6 vs. column 5, column 8 vs. column 7). The POI distribution reflects the functionality of a region. The most correlated POI major category is “professional”, under which there are a lot of venues like transportation center and conventional center. These are locations with more dynamic movements of

Table 2.3. Performance evaluation. Various feature combinations are shown in each column. The linear regression model and negative binomial results are compared by year group.

			Settings							
Column ID			1	2	3	4	5	6	7	8
Features[1]		**D**emo	✓	✓	✓	✓	✓	✓	✓	✓
		Geo					✓	✓	✓	✓
		POI		✓		✓		✓		✓
		Taxi			✓	✓			✓	✓
Year	Model[2]	Error								
2010	LR	MAE	394.41	416.98	408.09	406.93	394.78	432.45	402.25	416.41
		MRE	0.294	0.311	0.304	0.304	0.295	0.323	0.300	0.310
	NB	MAE	391.53	333.14	395.64	323.47	389.55	350.06	387.43	**320.75**
		MRE	0.292	0.249	0.295	0.241	0.290	0.261	0.289	**0.239**
2011	LR	MAE	380.22	409.30	396.97	401.11	379.61	422.94	389.39	408.91
		MRE	0.295	0.318	0.309	0.312	0.295	0.328	0.302	0.320
	NB	MAE	381.11	332.62	388.81	328.94	378.84	345.24	381.33	**335.97**
		MRE	0.296	0.259	0.302	0.256	0.294	0.268	0.296	**0.253**
2012	LR	MAE	378.91	412.95	401.54	412.20	376.53	423.88	399.25	419.93
		MRE	0.306	0.334	0.325	0.333	0.304	0.343	0.322	0.339
	NB	MAE	386.31	337.24	389.58	331.41	384.23	352.22	381.67	**345.49**
		MRE	0.312	0.273	0.315	0.268	0.310	0.284	0.308	**0.279**
2013	LR	MAE	367.89	420.81	390.75	402.75	369.24	433.48	388.92	412.31
		MRE	0.324	0.370	0.344	0.354	0.325	0.381	0.342	0.362
	NB	MAE	376.08	333.92	373.08	312.63	377.57	350.33	368.49	**319.86**
		MRE	0.331	0.294	0.328	0.275	0.332	0.308	0.324	**0.281**
2014	LR	MAE	331.28	375.53	349.00	350.31	329.93	386.90	345.79	361.28
		MRE	0.326	0.369	0.343	0.345	0.324	0.380	0.340	0.355
	NB	MAE	340.73	293.52	339.17	274.45	336.09	308.18	326.07	**273.27**
		MRE	0.335	0.289	0.334	0.270	0.331	0.303	0.321	**0.269**

[1] D – demographic features, G – geographical influence, P – POI features, T – taxi flow feature.
[2] LR – Linear Regression, NB – Negative Binomial Regression.

people. Such location information is not reflected in any of other features. POI thus provides unique information and it shows that using big data can benefit us in advancing the study of traditional crime inference problems.

Another issue that is worth discussing is whether POI is a surrogate of population features from demographics. That is, a region with POIs is a region with a higher population. However, as we see from Table 2.8, adding POI in addition to demographics always outperforms the features without POI. This is because

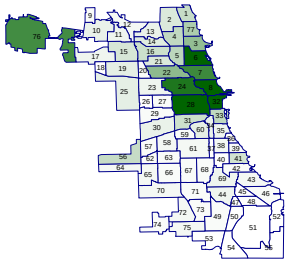


Figure 2.8. Absolute POI count distribution. In our crawled POI dataset, most community areas have less than 100 venues. Meanwhile, the downtown area there are over 10,000 venues for one community area, e.g. #8, #32.

population from demographics reflects the number of residents in that region, but POI reflects dynamics of population (e.g., people go to venues for food, entertainment, or travel). Therefore, the dynamic population in POI further complements the residential population in demographics.

2.5.2.3 Taxi Flow

The taxi flow is shown to improve the inference accuracy (see NB for column 3 vs. column 1, column 7 vs. column 5, column 8 vs. column 6). This validates our hypothesis that crimes do not only correlate with nearby regions but also correlate through hyperlinks on the space (i.e., the taxi flow).

Comparing column 7 (D+G+T) with column 5 (D+G), we find that the improvement by taxi flow is not obvious. However, comparing column 8 (D+G+P+T) with column 6 (D+G+P), we observe a much significant accuracy boost. The reason could be that the taxi flow further complements the POI data. When POI information is missing from the predictor, the city dynamics captured by taxi flow are weakened as well.

2.5.3 Feature Construction

There are different ways to use the POI and taxi datasets. In this section, we share our insights into the more effective ways in constructing the features.

2.5.3.1 POI Normalization

The straightforward definition of POI distribution is calculated by normalizing the POI count in each category by the total POI counts. However, the POIs in Chicago are not evenly distributed. As shown in Figure 2.8, most POIs are in the downtown area and some areas only have a few POIs. If normalized by the total number of POIs in a neighborhood, two neighborhoods may show similar distributions but they are quite different. For example, a downtown neighborhood and a distant neighborhood may both have a high ratio of the food category but the downtown neighborhood has many more POIs in total and is more dynamic in population constitution. Therefore, using the raw count instead of normalized distribution is more effective. This is also demonstrated in estimation accuracy as shown in Table 2.4.

Table 2.4. Using POI count instead of POI percentage improve the estimation accuracy. Estimation for crime in 2014 with all other features.

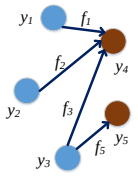
Scheme	NB	
	MAE	MRE
POI count	273.27	0.269
POI percentage	283.16	0.278

2.5.3.2 Taxi Flow Normalization

The taxi flow represents the interactions among community areas. There are several different approaches to incorporate the taxi flow into the model. First, we can use the raw taxi count as a weight on crime from other neighborhoods. One issue with the raw count is the concentration of taxi trips distribution in the downtown area. Consider the following example. In the downtown area, the average taxi flow count is 1000 between any pair of community areas, while the average of suburbs is 100. When we propagate crime by raw taxi count, the same amount of crime in downtown is propagated with a 10 times higher coefficient than that of suburb.

To address this issue, we can normalize the taxi flow, and there are two different approaches to normalize. 1) We can normalize the taxi flow by the total incoming traffic of the destination community area, and the semantics of this normalization is splitting the crime in the destination to all its neighbors. 2) Alternatively, we can normalize the taxi flow by the outgoing total trips in the source community

area. This normalization assumes the crime in each source community is spread out by the flow. The two normalization methods are shown in Figure 2.9.



Normalization by source:
$$F_4^t = \frac{f_1}{f_1} y_1 + \frac{f_2}{f_2} y_2 + \frac{f_3}{f_3+f_5} y_3$$

Normalization by destination:
$$F_4^t = \frac{f_1}{f_1+f_2+f_3} y_1 + \frac{f_2}{f_1+f_2+f_3} y_2 + \frac{f_3}{f_1+f_2+f_3} y_3$$

Figure 2.9. Two different normalization schemes.

Table 2.5. Various approaches to construct taxi flow feature. Estimation for crime in 2013 with all other features.

Settings	NB	
	MAE	MRE
Taxi flow count	368.71	0.324
Taxi flow normalized by source	349.38	0.307
Taxi flow normalized by destination	319.86	0.281

In Table 2.5 we compare the different approaches to handle the taxi flow. Using raw taxi flow count is clearly not a good option, due to the unbalanced data distribution. We also observe that normalizing taxi flow by destination is better than normalization by source. The reason could be explained by the example given in Figure 2.9. Suppose the focal region is a transportation hub, which has a lot of isolated regions connected to it. If we normalize the crime by source region, then the taxi flow feature of focal region is overestimated, since the coefficients of its neighbors do not sum to one.

2.5.4 Feature Importance

In this subsection, we study the importance of features through significance tests and coefficient changes over the years.

2.5.4.1 Significance Test

From previous results, we see that combining POI features and taxi flow will help improve the estimation accuracy. Now we try to measure the significance of this

accuracy boost by permutation tests. If a feature correlates with crime, when we randomly permute the values of this feature among neighborhoods, we will expect a higher error in crime estimation. So in each round of permutation, we can get an error in estimation. We compare the error with the original feature to the error distribution obtained from permutations. We conduct 1,000 rounds of permutations to approximately estimate the error distribution. The position of the original error in this distribution indicates the significance of this feature. For example, if the original error is smaller than 99% of the errors from the permutations, the p-value is 0.99.

Table 2.6. Estimated p-value for each feature. The p-value is defined as the possibility that a smaller error measure is observed under the null hypothesis.

Settings: D+S+P+T	LR		NB	
	MAE	MRE	MAE	MRE
	412.31	0.363	319.86	0.281
Feature	p-value			
D (demographics)	0.000	0.000	0.000	0.000
G (geographic inf.)	0.640	0.664	0.602	0.565
P (POI distribution)	0.025	0.025	0.001	0.001
T (taxi flow)	0.000	0.000	0.000	0.000

In Table 2.6, the p-values of different features are given. The demographics feature is the most significant with estimated p-value equals to 0.00. In all the $1,000$ random permutations of demographic feature, we never observe an error lower than the original error. The proposed POI distribution and taxi flow are significant as well, with a p-value of 0.5% and 1.3% for the negative binomial model. One interesting observation is that the geographical influence is not significant at all. One possible reason is that the demographics features capture the similarity of geographical neighbors, and therefore are surrogates of geographical influence.

2.5.4.2 Coefficient Study

In our regression model, the coefficient also indicates the importance of features. We normalize the values of all features to the range $[0, 1]$, so that coefficients are comparable. The top-6 features with the most significant coefficients are shown in Table 2.7. The top 3 rows in Table 2.7 are features with positive coefficients,

which implies the positive correlation with Crime. The three features with negative coefficients are negatively correlated with crime.

By comparing the coefficients over different years, we observe that the coefficients are relatively stable with respect to time. The most important feature is always POI professional category, which represents many populated public areas. The other two important demographic features are disadvantage index and percentage black. We also find three POI categories are among the top negatively correlated features. They are the residence, shop, and education categories. The reason is that at those places the population is relatively stable, which provides less opportunity for crime.

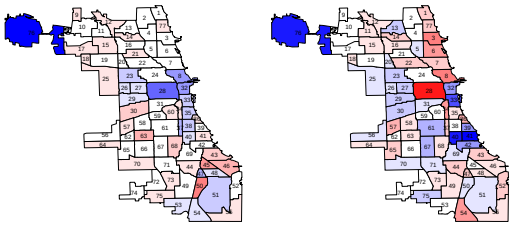
Table 2.7. The coefficients of the top-6 features over different years. There are 21 different features in total. Due to limited space, we only show the top 3 features with the highest positive/negative coefficients respectively.

Feature	Year				
	2010	2011	2012	2013	2014
POI professional	1.414	1.733	1.905	2.206	1.874
pct black	1.376	1.370	1.301	1.296	1.252
disadvantage index	1.237	1.055	1.270	1.700	1.462
POI education	-1.171	-1.265	-1.735	-2.041	-1.871
POI shops	-2.671	-2.747	-2.687	-2.549	-2.834
POI residence	-3.059	-2.719	-2.424	-2.151	-2.459

2.5.5 Improvements on Different Regions

The POI distributions are different from region to region. It is interesting to find out whether POI distribution is consistently positive in making the crime estimation better. We calculate the difference in estimation error (MAE) between two settings: 1) using demographics, geographical influence, taxi flow; and 2) using all these three features plus POI distribution. The similar measurement is calculated for the taxi flow feature. The results are shown in Figure 2.10. A positive difference (blue area) indicates that adding the new feature will help reduce the estimation error, while a negative difference (red area) indicates that the new feature adds more noise to the data.

It is interesting to find out that in the downtown area, i.e. community area #8, #32, #28, and #33, POI significantly improves the estimation accuracy. The



(a) POI (b) Taxi flow

Figure 2.10. Performance improvement per region by using POI or taxi flow features on 2014 crime. The difference of MAEs in estimating crime with/without POI feature is shown on the left, and the same measure of taxi flow is shown on the right. The color blue means the MAE is reduced by adding corresponding feature (i.e., better performance), while the red means the MAE is increased (i.e., worse performance). The color saturation indicates the value of difference.

reason is two fold. 1) The demographics information from census is mostly about the residing population in the focal area. However, in the downtown area there are a lot of floating population groups conducting various social activities, and this is not reflected by the census demographics. The POI information, on the other hand, reflects the functionality of a region, and plays a complementary role of demographic information. 2) In the downtown area, there are much more POIs than any other places, which provides more complete information about the community profile.

As for the taxi flow feature, it helps the most in those suburb area, because the taxi flow reflects the social interaction in those areas. In the downtown, the taxi flow feature incurs a relatively large estimation error. The reason is that the taxi flow distribution in Chicago is extremely skewed. Roughly 61% of the Chicago taxi trips have a destination in the downtown area, which may result in the model over-propagating crime estimates from all of Chicago into the downtown area.

2.6 Related Work

In the criminology literature researchers have studied the relationship between crime and various features. Examples are historical crime records [71, 72], education [73], ethnicity [74], income level [75], unemployment [76], and spatial proximity [48]. In data mining, newer types of data are used in the study. For example, there

are studies using twitter to predict crime [77, 78], and studies using cellphone data [59, 79] to evaluate crime and social theories at scale. Overall, the existing work on crime prediction can be categorized into three paradigms.

Time-centric paradigm. This line of work focuses on the temporal dimension of crime incidents. For example, in a study [71], the authors propose to use a self-exciting point process to model the crime and gain insights into the temporal trends in the rate of burglary. In another study [80], the authors investigate the temporal constraints on crime, and propose an offender travel and opportunity model. This paper validates the claim that a proportion of offending is driven by the availability of opportunities presented in the routine lives of offenders.

Place-centric paradigm. Most existing work adopt a place-centric paradigm, where the research question is to predict the location of crime incidents. The predicted crime location is usually referred by the term *hotspot*, which has various geographical size. There are plenty of studies on exploration of the crime hotspots. For example, in a study [81] the authors use criminal offense records to identify spatio-temporal patterns at multiple scales. They employ various quantitative tools from mathematics and physics and identify significant correlation in both space and time in the crime behavioral data. Short *et al.* [82] use a simple model to study the dynamics of crime hotspots and identify stable hotspots, where criminals are modeled as random walkers. Bogomolov *et al.* [59] use human behavioral data derived from mobile network and demographic sources, together with open crime data to predict crime hotspots. They compare various classifiers and find random forests have the best prediction performance. The paper [77] uses automatic semantic analysis to understand natural language Twitter posts from which the crime incidents are reported. Some other work [83, 84] employ kernel density estimation (KDE) to identify and analyze crime hotspots. Those studies form another form of crime prediction, which relies on the retrospective crime data to identify areas of high concentrations of crime. In [85], the authors extend the crime cluster analysis with a temporal dimension. They employ the space-time variants of KDE to simultaneously visualize geographical extent and duration of crime clusters.

Population-centric paradigm. In the last paradigm, research focuses on the criminal profiling at individual and community levels. At the individual level, [72] aim to automatically identify crimes committed by the same individual from a historical crime database. The proposed system, called *Series Finder*, is designed

Table 2.8. Crime rate inference results. Various feature combinations are shown in each column. The linear regression and negative binomial regression are compared by year group.

Features	**D**emo	✓	✓	✓	✓
	Geo	✓	✓	✓	✓
	POI		✓		✓
	Taxi			✓	✓
LR	MAE	329.93	386.90	345.79	361.28
	MRE	0.324	0.380	0.340	0.355
NB	MAE	336.09	308.18	326.07	**273.27**
	MRE	0.331	0.303	0.321	**0.269**

to find and classify modus operandi (M.O.) of criminals. At the community level, Buczak *et al.* [86] use fuzzy association rule mining to find crime patterns. The rules they found are consistent across all regions. The paper constructs association rules from population demographics in communities. In another paper [79], the authors use computational methods to validate various social theories at a large scale. They used mobile phone data in London, from which they mine the people dynamics as features to correlate with crime.

Our problem is different from the first two categories of work, mainly because our innovation lies in using newer type of data to enhance the commonly used traditional counterparts. More specifically, we use POI to enhance the demographics information and use taxi flow as hyperlinks to enhance the geographical proximity correlation. Although our problem does not consider the temporal dimension of crime in depth, it could be a promising supplement to better profile crime. Our problem does not predict the location of any particular crime incident. Therefore the methods proposed in place-centric methods are not applicable in our problem. However, the features we proposed may be incorporated in those crime prediction models.

Our problem falls into the third paradigm because we try to profile the crime rate for Chicago community areas. In our problem, the community areas are well-defined and stable geographical regions. The newly proposed POI features and taxi links providenew perspectives in profiling the crime rate across community areas.

2.7 Conclusion

In the social science literature, the demographics and geographical neighbors are known to exhibit strong correlations with crime. In this paper we solve the problem of crime rate inference with new features. More specifically, we propose to use POI features to assist the demographic features, and to use taxi flow as hyperlinks to supplement the geographical neighbors. The intuition behind the POI feature is that the POI distribution across community areas reflects profiles of the region functionality. The intuition behind the hyperlinks is that the taxi flow models the social interaction among nonadjacent regions, which potentially propagate crime or resources and information used in crime control. We adopt the negative binomial regression modal over the linear regression model, mainly because the count based regression models and guarantees positive prediction, while the linear regression may give negative crime rate as prediction. Both POI and taxi flow features from a publicly accessible dataset in Chicago are evaluated to be helpful. In the best scenario, the POI distribution and taxi flow reduces the prediction error by 17.6%.

Chapter 3
Region Representation Learning via Mobility Flow

3.1 Introduction

As of 2016, more than 54% of the world's population live in urban areas, and the percentage is expected to increase to 66 by 2050 [87]. In the meantime, increasing amount of urban data are being accumulated in the digital form, such as human traces, traffic, venues, local events, etc. Many cities (e.g., New York City, Chicago, and Los Angeles) have joined the open data initiative and created websites to release the city data to the public [2, 88]. Analyzing such data could provide us with valuable insights into our urban dynamics, and make the city smarter.

In this paper, we are interested in a typical inference problem, which aims to infer a regional property (i.e. crime rate, personal income, and real estate price) from observed auxiliary urban features. Such inference can help us better understand the correlations among urban properties. Recent work [89] has shown that the taxi volume could be used as a similarity measure between regions. Through the similarity measure from traffic flows, we are able to employ the target variable of relevant regions to improve the predication accuracy. The intuition is that a large volume of flow from region A to region B indicates that the properties of A and B should be more relevant, and thus we could use one to predict the other. Although the intuition seems straightforward, there are some issues with it. Consider the example in Figure 3.1. On the left, there is no significant flow between two solid blue circles. However, since they share a lot of common neighbors, it is reasonable

to assume they are relevant. On the right, the solid blue circle is a hub with strong connections to other regions. However, the hub could be a downtown area and play a different role compared with other regions.

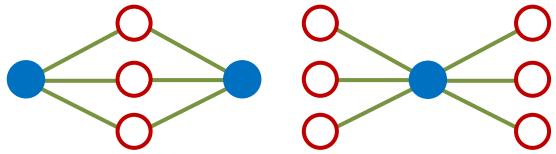


Figure 3.1. Each node is a region. The edge represents a significant amount of taxi flow between two regions.

The example above motivates us to account for the structural information of mobility flow graph. Graph embedding method [90, 91] can be one possible solution to model such structural information. A good region representation learned from mobility flow graph may help us better capture the relationships between regions and thus the correlations can be used to improve the inference model.

However, utilizing taxi flow data to learn the representations of regions is non-trivial. In literature, a transition matrix has been frequently used to represent mobility flow data [92–95]. In this transition matrix, a region i can be represented by an n-dimensional vector, where the j-th element in the vector indicates the traffic volume from region i to j (out flow) or from region j to region i (in flow). However, such a representation does not consider the temporal dynamics. For example, the flow volume from region A to downtown might be the same as that from region B to downtown. Without considering the temporal dynamics, we cannot differentiate A and B w.r.t. downtown. However, the flow from A to downtown might mostly be morning transitions whereas the flow from B to downtown happens in the evenings. Downtown region might function as a working place for people living in Region A but function as an after-work entertainment region for people working in region B. So A and B should be different in their presentations since they have different relationships with downtown region.

To consider the temporal dynamics, we could construct a tensor by adding a time dimension in addition to the transition matrix [28]. However, such a tensor does not capture the multi-hop transitions between regions. For example, there could be strong flow from residential area A to working area B in the morning, and then from working area B to restaurant area C in the evening. This indirect transition relationship between the region A and C cannot be captured in the

pairwise transition matrix.

We propose to learn representations of regions by adapting recent embedding techniques, which have demonstrated successful results in word embedding [96–99] and graph embedding [90, 91, 100]. However, the mobility flow data input are quite different from those data. The key challenge lies in how to generate a meaningful context for a region using the mobility flow data, in a similar way to the sentence context for word embedding or the neighbor context for graph embedding. Another challenge lies in the data sparsity. Even though we have a huge mobility dataset, the data follow a long-tail distribution w.r.t. regions. We could still have no information for some remote regions at certain times such as midnight, and thus it is difficult to learn their corresponding representations.

We propose a region embedding method by considering both temporal dynamics and multi-hop transitions. We define a flow graph, where each vertex represents one region within a certain time interval and edges represent the transition flow between two regions at different time intervals. The structure encodes both temporal dynamics and multi-hop transitions. To further address the sparsity issue, we define a spatial graph, and learn the embedding jointly on the flow graph and the spatial graph. The spatial graph captures the geographical adjacency among regions and complements the flow graph.

In experiment, we evaluate our embedding methods on three prediction tasks. The proposed embedding method is shown to consistently outperform existing methods. In order to interpret the semantic meaning of learned embeddings, we conduct a quantitative analysis on taxi and POI data, and also give a case study on the embedding visualizations.

To summarize, the key contributions of this paper are:

- We study a generalized inference problem in the urban setting. We propose to learn region embedding from mobility flow data to enhance the inference model.
- In order to incorporate both temporal dynamics and multi-hop transition and also to address the sparsity issue, we propose to jointly learn the embedding from the flow graph and the spatial graph.
- We validate our method through extensive experiments on multiple datasets.

The rest of this paper is organized as follows. Section 3.2 gives the motivation of learning region embeddings. Section 3.3 presents the formal definition of our

problem. Section 4.4 introduces the dynamic embedding method in detail. The experiment results are given in Section 3.5. Section 3.6 summarizes related work. Finally, we conclude in Section 4.6.

3.2 Preliminary

In this section we first present the generalized inference model, and then introduce our empirical observations on the relationship strengths measure.

3.2.1 Generalized Inference Model

The generalized inference model is a typical regression task to study the urban dynamics from various data sources. Given a set of K non-overlapping regions $\mathcal{R} = \{r_1, r_2, \cdots, r_K\}$, we are interested in estimating the target variable for every region, denoted as y_i for region r_i. We only have observations of target variables on a subset of regions. However, we observe some auxiliary features for all of the regions, such as the demographics and average income. These auxiliary features are denoted as $X_i \in \mathcal{R}^d$ for region r_i, where d is the dimension of auxiliary features.

To predict the target variable, we use the following generalized regression model

$$y_i = \alpha \cdot X_i + \beta \sum_{j \in \mathcal{N}_i} sim(i, j) \cdot y_j + \gamma, \tag{3.1}$$

where α, β, and γ are parameters of the regression model. The term $\sum_{j \in \mathcal{N}_i} sim(i, j) \cdot y_j$ accounts for the propagation effect of neighboring regions of r_i, where $\mathcal{N}_i$ is a set of neighboring regions of r_i and $sim(i, j)$ measures the relevance of region pair $\langle r_i, r_j \rangle$. The relevance function $sim(i, j)$ is usually defined with extra information, such as the spatial information.

It is the relevance function $sim(i, j)$ that allows us to generalize the base model. For example, a straightforward definition with hard boundary is $sim(i, j) = \begin{cases} 1 & \text{if } r_i \in \mathcal{N}_j, \\ 0 & \text{otherwise,} \end{cases}$ where $\mathcal{N}_j$ is the set of k-nearest neighbors to region r_j. In order to provide a more flexible way to control the relevance of neighbors, a soft version of the relevance function could be defined with a spatial distance measure as $sim(i, j) = \frac{1}{d(i,j)}$, where $d(i, j)$ is the distance between the centroids of two regions. The intuition is that the closer two regions are, the more relevant they are.

As newer type of data is available, such as the taxi commuting data, the relevance measure can be defined accordingly as $sim(i,j) = \frac{f(j,i)}{\sum_{p \in \mathcal{N}_i} f(p,i)}$, where $f(j,i)$ is the amount of flow from r_j to r_i. Recent work from Wang et al. [89] employs this definition and brings taxi flow feature into the model. The prediction model becomes

$$y_i = \alpha \cdot X_i + \beta_1 \vec{w_g}^T \vec{y} + \beta_2 \vec{w_f}^T \vec{y} + \gamma, \tag{3.2}$$

where $\vec{w_g}$ is the reverse distance weighting vector, $\vec{w_f}$ is a weighted taxi flow vector, and $\vec{y}$ is the target variables of all other regions. Moreover, Wang el al. [89] verifies that negative binomial regression is preferable to linear regression for predicting non-negative y_i. In the rest of this paper, we use the negative binomial regression as our generalized model, i.e.

$$y_i = \exp(\alpha \cdot X_i + \beta \sum_{j \in \mathcal{N}_i} sim(i,j) \cdot y_j + \gamma). \tag{3.3}$$

3.2.2 Empirical Study with Urban Data

To verify the taxi flow can serve as a good relevance measure, we first make some observations with crime data and taxi data in Chicago. For every pair of regions $\langle r_i, r_j \rangle$, we plot their crime rate differences against the flow volume $f(i,j)$ in Figure 3.2.

Overall, the blue points in Figure 3.2 validate the intuition of adding taxi flow into prediction model in Equation (3.2). However, we notice that there are many red region pairs do not follow this intuition. The reason is that the downtown region is contained in those pairs. Chicago downtown region has the highest crime rate, and there is a significant amount of traffic between downtown and other regions.

The observation above motivates us to look beyond the traffic volume to determine the relevance measure. As shown in the Figure 3.1, if we account for the structural information of mobility flow, the downtown is a popular hub, which differentiates itself from most of other regions. The graph embedding method is therefore a sound solution to estimate region relevance by modeling such structural information.

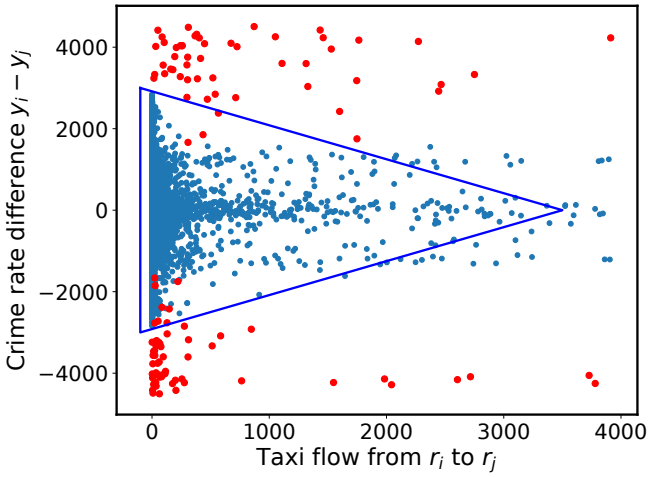


Figure 3.2. The crime rate difference vs. traffic flow volumes for every pair of regions $\langle r_i, r_j \rangle$. Points forming the blue triangle shape indicate that the larger the flow between region r_i and region r_j is, the difference between their crime rates is smaller. The red point denotes a pair of regions with one region being the downtown area.

3.3 Problem Definition

We define the region representation learning problem as a joint embedding learning problem on two different graphs — flow graph and spatial graph.

The input data consist of mobility data and spatial information of the city. The mobility dataset containing n trips is denoted as $\Gamma = \{\gamma^i\}$. Each trip γ has the format $\langle l_s, l_e, t_s, t_e \rangle$, where l_s and l_e are the starting and ending location coordinates (i.e., latitude and longitude), and t_s and t_e are the starting and ending time of the trip respectively. The spatial information of the city is a set of K non-overlapping regions, denoted as $\mathcal{R} = \{r_1, r_2, \cdots, r_K\}$. The regions could be defined as the administrative boundaries (e.g. tracts and community areas) or partitioned by the road network [28]. The spatial boundary of each region r_i is given as well. To simplify the temporal dynamics, we use relative timestamps within one day $\mathcal{T} = \{1, 2, \cdots T\}$ with fixed time intervals, such as 1 hour.

The same region at different timestamps bears different functions, and thus the embedding could be different. We differentiate the same region at different timestamps with different vertices in our heterogeneous dynamic graph with *time-enhanced vertex*.

Definition 1 (Time-enhanced vertex)**.** *Each vertex in our graph is denoted as* v_i^t,

which represents the region r_i at time t. We call this time-enhanced vertex *and our method will learn embedding representation for each such vertex. The set of time-enhanced vertices is denoted as $\mathcal{V}$, which contains $K \cdot T$ vertices, where K is the number of regions, and T is the number of timestamps.*

Given a set of vertices, there are two kinds of relations we want to capture. The first type of relation is derived from the mobility flow among those regions, which is formulated into the *flow graph* G_f. The second one is the spatial adjacency, which is defined by the *spatial graph* G_s. The intuitions and definitions of these two graphs will be introduced in detail in Section 4.4.

Our method learns the representations from both graphs simultaneously. The formal definition of our problem is as follows.

Definition 2 (Dynamic mobility graph embedding)**.** *Given the flow graph G_f and spatial graph G_s, we aim to learn a vector representation $\mathbf{u}_i^t \in \mathbb{R}^d$ in a low dimensional space for each vertex $v_i^t \in \mathcal{V}$, i.e. learning a mapping $f : \mathcal{V} \to \mathbb{R}^d$. In the d-dimensional embedding space, both the mobility flow relation and spatial adjacency are preserved.*

With the region embeddings, we define the relevance measure by their dot product, i.e. $sim(i, j) = \mathbf{u}_i^T \mathbf{u}_j$.

3.4 Method

In this section, we give the design motivations and formal definitions of the flow graph and spatial graph. Following the graph definitions, we describe the embedding learning objective. At last, we present the optimization techniques to learn the embedding.

3.4.1 Flow Graph

The same region at different time may carry different functions. Take the downtown area as example, which has mixed point-of-interests distribution. In the morning, people go to downtown mostly for work. Therefore, in the morning the downtown area acts as a professional area. However, at night there are also a significant

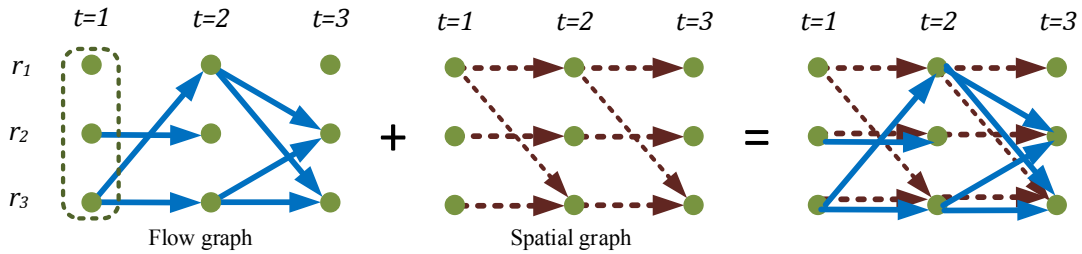


Figure 3.3. The layered structure of a flow graph (left), a spatial graph (middle), and the combined graph (right). Each row r_k is the region. Each column t is the timestamp, and all vertices within at the same timestamp (the dotted rectangle) form one *layer* of the graph. Each vertex v_i^t is a *time-enhanced vertex* refers to region r_i at time t. On the left, the solid blue edge refers to the taxi flow, and edge weight is number of taxi trips. In the middle, the dotted red edge refers to the spatial adjacency, and the edge weight is inversely correlated with the distance between region centroids. From the flow graph, vertices v_1^2 and v_3^2 have similar embeddings because they have similar in-flow from v_3^1 and similar out-flow to v_2^3 and v_3^3. However, with flow graph alone, we are not able to learn the embeddings for v_1^1 and v_1^3, due to lack of traffic flow. The spatial graph provides spatial information, which makes it possible to learn an embeddings for v_1^1 and v_1^3.

amount of people travel to downtown for food and drink, and the downtown acts as an entertainment area.

The aforementioned example motivates us to learn different embeddings for the same region at different times. Follow this intuition, we design the flow graph G_f as a layered graph, shown on the left of Figure 3.3, and formally defined as follows.

Definition 3 (Flow graph)**.** *The* flow graph *is a layered graph defined as* $G_f = (\mathcal{V}, E_f)$*. The vertices* $\mathcal{V} = \{v_i^t\}$ *is the set of time-enhanced vertex. The vertices with the same timestamp are grouped into one* layer*, and there are* T *layers in total. The edge set* E_f *only contains one type of edges* $\{e_{ij}^t\}$*, where* e_{ij}^t *connects vertices* v_i^t *and* v_j^{t+1} *from two consecutive layers. The edge weight* f_{ij}^t *is the volume of mobility flow.*

The flow graph models the mobility pattern of crowd in the city. More specifically, we can sample a lot of paths from the flow graph to represent human trajectories. Each path consists of a sequence of regions, whose timestamps are monotonically increasing with a fixed step of 1. The length of each path is bounded by the number of timestamps in the graph. And each path semantically refers to one trajectory of a individual. For example, one possible path is ⟨home, 8:00 am⟩ →

⟨office, 9:00 am⟩ → ⋯ → ⟨office, 6:00 pm⟩ → ⟨bar, 7:00 pm⟩.

However, there are three issues with a path sampled from the flow graph. First, the flow graph does not deal with the fact that people travel to a region and stay there. For example, in the left-most graph in Figure 3.3, the edge between v_1^1 (node of region r_1 at time $t = 1$) and v_1^2 (node of region r_1 at time $t = 2$) is missing, which means there is no trip observed transiting within the same region, but there could be people staying in that region. Second, the flow graph suffers data sparsity issue. If there is no traffic flow going in/out some region during a time slot, then it is impossible to learn the embedding of this region at that time. Third, the flow graph cannot recognize the same or nearby region across different time slots. More specifically, the flow graph treats all $K \cdot T$ vertices as independent regions. However, it is very likely that the same region in different time slots are strongly correlated. Recall the residential area example, where the large volume of in-going flow at night is caused by the large volume of out-going flow in the morning.

3.4.2 Spatial graph

To address the issues with the flow graph, we propose a spatial graph, which is defined as:

Definition 4 (Spatial graph)**.** *The* spatial graph *is a layered graph as well, denoted as $G_s = (\mathcal{V}, E_s)$. The vertices set $\mathcal{V}$ is exactly the same as that of flow graph. The edge set E_s also only contains edges connecting two vertices from consecutive layers. The edge weight g_{ij}^t represent the spatial similarity of two regions, which is inversely correlated with distance.*

The spatial graph share the same structure and exactly the same vertices with the flow graph. The only difference is that the edges in spatial graph are constructed differently. The basic assumption behind the spatial graph is that human mobility are bounded by space. When there is no transition observed, the probability that people appeared at a different region is inversely correlated to the distance they need to travel. Therefore, two regions that are close in space should have stronger correlation in their embeddings. In spatial graph, the edges e_{ij} refers to the spatial similarity between regions r_i and r_j. The edge weight g_{ij} is inversely correlated

with the distance, formally defined with exponential decay function [101] as follows

$$g_{ij} = \exp(-C \cdot d_{ij}), \tag{3.4}$$

where d_{ij} is the spatial distance between the centroids of two regions. We should notice that the spatial graph is static over time, therefore, all edges between any two consecutive layers are actually the same. C is a parameter controls the exponential decay rate of the distance. Larger C means faster decay, which makes regions far away has little correlation with current region.

The design of spatial graph, shown in the middle of Figure 3.3, naturally incorporates the spatial adjacency. This spatial adjacency could be regarded as a transition cost, which helps us to estimate the stay probability. Even more, the spatial adjacency enables the embedding learning for regions without any taxi flow, which solves the sparsity issue. Lastly, the spatial adjacency identifies the same region across different timestamps, because the edge weight between a pair of time-enhanced vertices on the same region is always the maximum edge weight.

3.4.3 Heterogeneous Graph Property

Combine the flow graph and spatial graph together, we get a heterogeneous graph that represents the crowd mobility pattern on the right of Figure 3.3. In this heterogeneous graph, one path convey more information about crowd mobility pattern than one path from the flow graph. Now it is possible for a path to capture both transition and stay, such as $\langle$home, 8:00 am$\rangle \rightarrow \langle$office, 9:00 am$\rangle \xrightarrow{\textbf{stay}}$ $\langle$**office, 10:00am**$\rangle \rightarrow \cdots \rightarrow \langle$office, 6:00 pm$\rangle \rightarrow \langle$bar, 7:00 pm$\rangle$.

This heterogeneous graph has two properties that fit the requirement of our problem. (1) The graph is still a temporal graph, which enables us to learn a dynamic embedding for each region. (2) In the heterogeneous graph, the multi-hop temporal dependency is captured within each path. The multi-hop temporal dependency is important to differentiate region functions. For example, at 6:00 pm we observe same amount of flow going into region A and B, which makes it difficult to differentiate the function of A and B. But if we know that in the morning, there is a large amount of flow going out of A, while almost no flow going out of B, then A is more likely to be a residential area, whereas B is more likely to be an after-work entertainment region.

3.4.4 Embedding Learning Objective

In order to capture two properties mentioned above, we propose to use the embedding technique to learn the representation of each region. The reason is that graph embedding explicitly captures the multi-hop dependency. Meanwhile, the baseline method for graph representation learning, such as directly using the in/out flow as vector representation or matrix factorization, is not able to capture the multi-hop correlation.

3.4.4.1 On Single Graph

The embedding learning process on the flow graph and the spatial graph are exactly the same, due to the fact that both graphs have similar structure. Without loss of generality, we take the flow graph as example to explain the learning process.

First we define a path as $P_i = v_{i_1} v_{i_2} \cdots v_{i_m}$, whose starting and ending vertices are v_{i_1} and v_{i_m} respectively. We omit the time superscript, because the time slots for the vertices of path P must be monotonically increasing with fixed step size 1. And we denote the relation that a path contains a vertex v_i^t as $v_i^t \in P$. With the definition of path, we further define the set of paths containing v_i^t as $\mathcal{P}(v_i^t) = \{P_i | v_i^t \in P_i\}$. The context of one vertex v_i^t, which refers to all the other vertices that are multi-hop neighbors of v_i^t, is defined as $C(v_i^t) = \{v_c | \exists P_i \in \mathcal{P}(v_i^t), v_c \in P_i\} \setminus \{v_i^t\}$.

We adopt the skip-gram model [97] to learn the embedding $\mathbf{u}_i^t$ for each node v_i^t. Formally, we estimate

$$p_f(v_c|v_i^t) = \frac{\exp(\mathbf{u}_i^{t^T} \mathbf{u}_c)}{\sum_{v_{i^*} \in C(v_i^t)} \exp(\mathbf{u}_i^{t^T} \mathbf{u}_{i^*})}, \tag{3.5}$$

where v_c is one vertex in v_i^t's context $C(v_i^t)$, $\mathbf{u}_c$ and $\mathbf{u}_i^t$ are the embeddings of v_c and v_i^t respectively.

The empirical conditional probability $\hat{p_f}(v_c|v_i^t)$ is estimated by the volume of mobility flow in the flow graph. More specifically, if v_c is within the context of v_i^t, there must be at least one path from v_c to v_i^t or a path from v_i^t to v_c. Without loss of generality, we assume one of the path is from v_i^t to v_c with m vertices, denoted as P_i, where $v_{i_1} = v_i^t$ and $v_{i_m} = v_c$. First we estimate the transition probability of two adjacent vertices. Then the empirical probability $\hat{p_f}(v_c|v_i^t)$ is estimated from this transition probability.

The transition probability between two directly connected vertices $v_{i_k}^t$ and $v_{i_{k+1}}^{t+1}$ is given by

$$p(v_{i_{k+1}}^{t+1}|v_{i_k}^t) = \frac{f_{i_k i_{k+1}}^t}{\sum_{v_{j^*} \in N(v_i^t)} f_{i_k j^*}^t}, \tag{3.6}$$

where $N(v_i^t)$ refers to the direct next-hop neighbors of vertex v_i^t, and $f_{i_k i_{k+1}}^t$ refers to the weight of edge $e_{i_k i_{k+1}}^t$ in the flow graph. Therefore, the transition probability from v_{i_1} to v_{i_m} through P_i is

$$\begin{aligned} p(P|v_{i_1}) &= p(v_{i_m}, v_{i_{m-1}}, \cdots, v_{i_2}|v_{i_1}) \\ &= \prod_k^m p(v_{i_k}|v_{i_{k-1}}, v_{i_{k-2}}, \cdots, v_{i_1}) \end{aligned} \tag{3.7}$$

Due to the Markov property, Equation (3.7) becomes

$$p(P|v_{i_1}) = \prod_k^m p(v_{i_k}|v_{i_{k-1}}), \tag{3.8}$$

The empirical conditional probability $\hat{p_f}(v_c|v_i^t)$ is

$$\hat{p_f}(v_c|v_i^t) = \sum_{P_i \in \mathbb{P}} p(P_i|v_i^t), \tag{3.9}$$

where $\mathbb{P}$ is the set of all paths starting at v_i^t and ending at v_c. Finally, we can learn the embedding by minimizing the difference between two distributions $p_f(v_c|v_i^t)$ and $\hat{p_f}(v_c|v_i^t)$. The objective is

$$O_f = D(p_f(\cdot|\cdot), \hat{p_f}(\cdot|\cdot)), \tag{3.10}$$

where D is the distance function for two distributions, and one commonly used function could be the KL divergence.

The embedding learning objective of spatial graph is similar to the flow graph. We minimize the difference between the embedding distribution and empirical distribution, which is

$$O_s = D(p_s(\cdot|\cdot), \hat{p_s}(\cdot|\cdot)). \tag{3.11}$$

3.4.4.2 On Heterogeneous Graph

In order to learn our embedding on two graphs simultaneously, we combine Equation (3.10) and Equation (3.11), and the joint learning objective is

$$O = O_f + O_s = D(p_f(\cdot|\cdot), \hat{p_f}(\cdot|\cdot)) + D(p_s(\cdot|\cdot), \hat{p_s}(\cdot|\cdot)). \tag{3.12}$$

3.4.5 Embedding Learning Optimization

3.4.5.1 On Single Graph

Directly optimizing the objective in Equation (3.10) and Equation (3.11) is computationally expensive, due to two reasons.

1. To calculate the conditional probability $p_f(\cdot|v_i^t)$ in Equation (3.5), for each v_i^t it requires the summation over the entire set of vertices. Therefore, the overall complexity is $O(K^2 \cdot T^2)$, where $K \cdot T$ is number of vertices.

2. To estimate the empirical conditional probability $\hat{p_f}(v_c|v_i^t)$ in Equation (3.9), for every pair of vertices we have to sum over all paths P_i among them, which is exponential to the number of vertices in one layer.

To address the first problem, we adopt the negative sampling approach proposed in [96], which samples multiple negative pairs from a noise distribution to estimate one true pair. The objective is given by

$$\log \sigma(\mathbf{u}_i^{t^T} \mathbf{u}_c) + \sum_q^s \mathbb{E}_{v_q \sim P_n(v_i^t)} \left[\log \sigma(-\mathbf{u}_q^T \mathbf{u}_c) \right], \tag{3.13}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function, $P_n(v_i^t)$ is the noise distribution, and s is the number of negative samples. The Equation (3.13) is used to replace every $\log p(v_c|v_i^t)$ term during the skip-gram optimization.

To address the second problem, we use the graph sampling method to estimate the empirical probability $\hat{p}(v_c|v_i^t)$. More specifically, we generate m paths from the graph via random walk. Due to the special structure of our layered graph, the time index of the sequence must be monotonically increasing with fixed step size 1. Given that m is large enough, we could use the co-occur frequency count from those random walks to estimate $\hat{p}(v_c|v_i^t)$ with sufficient accuracy.

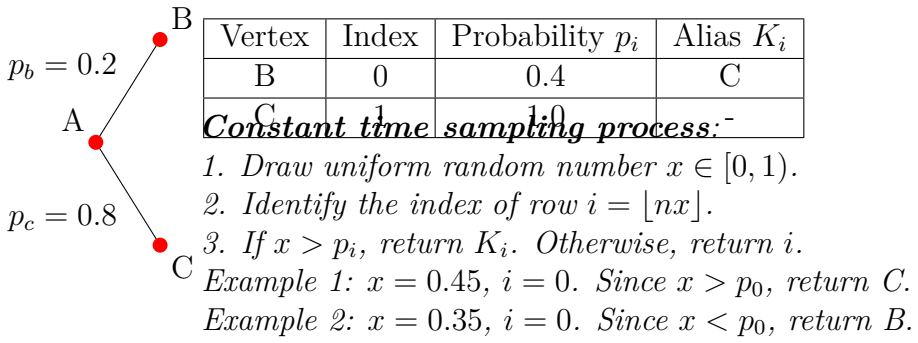


Figure 3.4. The alias method explanation. On the left, we want to draw the next vertices of A. The probability table and alias table are created on the top right. The bottom right shows the constant time sampling process from the alias method.

The random walk boils down to next-vertex sampling according to the edge weights. Since the random walk is conducted on a weighted graph, at each vertex, we sample the next vertex according to the out-degree distribution, which could be expensive. The straightforward method is to convert each weighted edge into an interval within the range of $[0, w_{sum})$, where the w_{sum} is the sum of out degree at current vertex. The sampling process is that first generate a uniform random number $x \in [0, w_{sum})$, and then find the interval that x maps into. Therefore, this next-vertex sampling method takes $O(K)$ time, where K is the number of vertices in one layer, which is also the upper bound of number of outgoing edges from current vertex. Since we are generating m paths, we have to conduct $m \cdot T$ next-vertex sample process, where T is the upper-bound of the path length. The overall complexity would be $O(m \cdot T \cdot K)$.

We further boost the next-vertex sampling process with alias method [102]. The advantage of alias method is that it is possible to repeatedly sample next edge with constant time, after preprocessing outgoing edges and save the information. More specifically, the alias method creates two tables for the next edges as shown in Figure (3.4). The alias method makes the path sampling significantly faster, because in our path sampling process we repeatedly sample on each vertex. For each vertex, the initialization of alias tables take $O(K)$, where K is the upper bound for the number of next vertex. Therefore, the overall initialization takes $O(K \cdot T \cdot K) = O(K^2 \cdot T)$, where $K \cdot T$ is the number of alias tables need to create. The overall sampling process takes $O(m \cdot T)$. Since $m \gg K$, it is safe to assume that $m > K^2$, and then the overall complexity is $O(K^2 \cdot T + m \cdot T) = O(m \cdot T)$.

The experimental comparison of alias method with the simple method is described in Section 3.5.2.3.

3.4.5.2 On Heterogeneous Graph

The sampling-based method above can be easily applied on the heterogeneous graphs to learn the joint embedding. We conduct random walk on both graphs to generate path simultaneously. Then we feed all the paths to the skip-gram neural networks model to learn a joint embedding for each vertex.

3.4.6 Discussion: Path Sampling

Here we draw a connection between our graph sampling-based optimization technique and word2vec [97] in language modeling method. The goal of word2vec is to build vector representations of words using probabilistic neural networks. This idea could be re-purposed to model the graph structure as well [90], due to the power law property in both the degree distribution in a graph and the word frequency distribution in natural language.

We regard the set of vertices in the graph as a special corpus, and each vertex is a word. The path sampled from the weighted graph via random walk can be thought of sentences. The multi-hop neighbors of a vertex in the path is similar to the word context. Therefore, estimating the neighboring vertices of a given vertex is analogy to the skip-gram language model [100].

3.5 Experiment

In this section, we first describe datasets and experiment settings. Then we evaluate the effectiveness and efficiency of proposed embedding method with several prediction tasks. Finally, we interpret the semantic meaning of the learned embedding with both quantitative analysis and case study.

3.5.1 Settings

3.5.1.1 Data description.

We study the urban dynamics at community area (CA) level. A community area is a predefined administrative area in the city of Chicago. The geographic boundary information is available through US census survey [103]. The following urban data are collected and used in our evaluation.

Demographics data at community area level is made public by the US census bureau [103]. The demographic features mainly cover the following aspects of a community area: total population, population density, poverty, residential stability, and ethnic diversity.

Point-Of-Interest (POI) data is obtained through Foursquare API [104]. It contains more than $112,000$ POI records for Chicago. Each POI record provides venue name, category, number of check-ins, and number of unique visitors. We use the POI category distribution information of each region to measure the region functions. There are 10 major POI categories including arts & entertainment, education, event, food, nightlife, outdoor & recreation, professional, residence, shops and travel.

Taxi data [88] in Chicago from 2013 to 2015 are used to construct the mobility flow graph. There are over 86 million taxi trips recorded over the three years, which is roughly 2.4 million trips per month. For each trip, we have the following information available: pick-up and drop-off dates and locations. Due to privacy concern, in this dataset, all timestamps are rounded to closest 15 minute marks, and all locations are mapped to the center of census tracts.

Crime data is publicly available on Chicago Data Portal [105], which contains more than 5 million crime incidents from 2001 to current day. The incident date, location, and primary type of each crime incident are recorded.

House price data is obtained from Zillow real estate website [106]. We collect the sale price, floor size, latitude, and longitude information for over $45,000$ real estates that were sold within 2 years in the city of Chicago.

In order to evaluate and interpret our embedding results, we predict the following three target variables for each community area.

- Crime rate, which is crime incidents count per 10,000 population.

- Average personal income in dollar.
- Average house price with a unit of dollar per square foot.

3.5.1.2 Methods for comparison

For each prediction task, we follow the generalized regression framework in Equation (3.3). We use the state-of-the-art method in [89] as a base model, which does not employ the embedding technique to calculate relevances. Since the base model directly employs the traffic volume and inverse spatial distance as relevance measure, we denote it RAW in the rest of experiments.

We name our embedding method as **heterogeneous dynamic graph embedding ($HDGE$)**. This proposed dynamic embedding technique also applies to single flow graph or spatial graph, which are called DGE_{flow} and $DGE_{spatial}$ respectively. We set the embedding dimension as 8 for all methods. We compare $HDGE$ with two alternative embedding methods. First, we introduce a straightforward baseline approach for flow graph modeling, called *slotted graph*. Similar to flow graph, the slotted graph also accounts for the temporal dynamics. However, the slotted graph models the mobility flow for each time slot independently.

- **Matrix factorization (MF)** is a conventional method for dimension reduction. In order to get dynamic vector representations, the matrix factorization method is used to decompose the adjacency matrices of slotted graphs.
- **LINE** [91] is a graph embedding method that learns embedding on a weighted graph to encode both first and second order proximity. Applying LINE on the slotted flow graph also leads to an alternative temporal embedding.

3.5.1.3 Evaluation metrics

The dynamic embedding method learns different embeddings for different time slot. Within each time slot, we use leaned embeddings to calculate the relevance measures and evaluate the regression model with leave one out setting. The model performance is evaluated by mean relative error and mean absolute error:

$$MRE = \frac{1}{T}\sum_{t=1}^{T}\frac{\sum_i^n |y_{it} - \hat{y_{it}}|}{\sum_i^n y_{it}} \quad MAE = \frac{1}{T}\sum_{t=1}^{T}\sum_{i}^{n}|y_{it} - \hat{y_{it}}|,$$

where y_{it} is the ground truth value for target variable of region i at time slot t, and $\hat{y_{it}}$ is the estimate.

It is worthy mentioning that among all three target variables only crime rate presents daily periodicity. For average personal income and real estate price, the value of the same region does not change within one day, i.e. $\forall t \in \mathcal{T}$, $y_{it} = y_i$.

3.5.2 Evaluations

3.5.2.1 Feature Selection

For each predication task, we have four types of features available, which are demographic features (D), POI features (P), geographical feature (G), and taxi flow feature (T). In this section, we aim to identify the best feature combinations for each prediction task. We use the base model *RAW* for this purpose.

Table 3.1. Crime rate prediction with *RAW* from 2013 to 2015. The MAE unit is crime count per 10,000 population.

Year	2013		2014		2015	
Features[1]	MAE	MRE	MAE	MRE	MAE	MRE
D+P	15.03	0.318	13.26	0.317	7.31	0.335
D+P+G	15.54	0.329	13.75	0.326	7.46	0.337
D+P+T	14.52	0.308	12.79	0.307	7.15	0.322
D+P+G+T	14.92	0.316	13.15	0.316	7.35	0.332

[1] D – demographic features, G – geographical influence, P – POI features, T – taxi flow feature.

The crime rate prediction results of *RAW* method with different feature combinations are shown in Table 3.1. We only show the prediction results from year 2013 to 2015, because only in those years we have both taxi flow data and crime incident data. From Table 3.1, we observe that the best crime rate prediction is achieved by using only three types of features, i.e. demographics, POI, and taxi flow. Adding geographic features does not improve the prediction accuracy. This observation is actually consistent with previous work [89].

The average personal income and house price prediction results of *RAW* are shown in Figure 3.2. When making income prediction, we eliminate related features from the demographics features. To make fair evaluation, we try our best to align the time window of features and target variables. More specifically, the income census data is collected in 2010, and we use taxi flow in the closest year as features.

Table 3.2. Average personal income and house price prediction with *RAW*. The MAE unit of personal income is dollar. The MAE unit of house price is dollar per square foot.

Data	Income		House Price	
Features	MAE	MRE	MAE	MRE
D+P	15304	0.253	39.87	0.233
D+P+G	16905	0.279	41.40	0.242
D+P+T	15433	0.255	39.28	0.229
D+P+G+T	15127	0.250	40.728	0.238

The house price data is from 2015 to 2017, and the taxi flow in 2015 is used to predict house price.

From Table 3.2, we observe that the best feature combination for income prediction is to involve all four types of features. Meanwhile, the best feature combination for house price prediction is demographics, POI, and taxi flow.

In all three prediction tasks, the taxi flow features are consistently proven to effectively improve the prediction accuracy.

3.5.2.2 Embedding Evaluation

In this section, we evaluate the embedding results by calculating the relevance measures with learned embeddings. Without loss of generality, we define the relevance measure in Equation (3.3) by their dot product, i.e. $sim(i, j) = \mathbf{u}_i^T \mathbf{u}_j$.

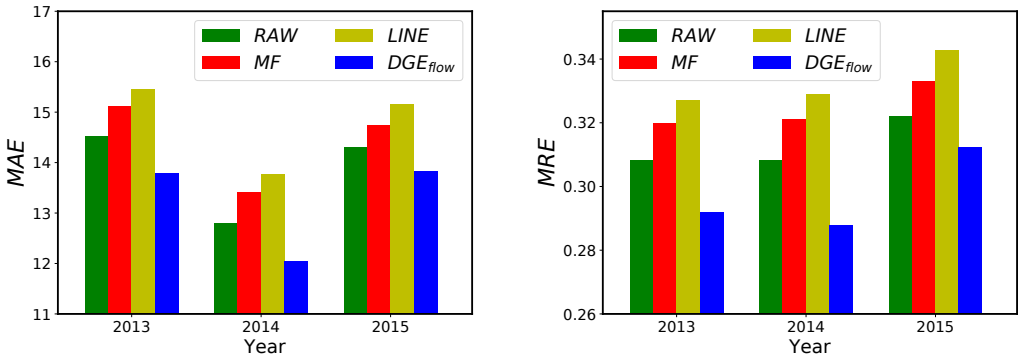


Figure 3.5. Crime rate prediction MAE (left) and MRE (right) with dynamic mobility flow embeddings.

The MAE and MRE of crime rate prediction in different years are shown in Figure 3.5. All methods use D+P+T feature combinations, and the MRE of *RAW*

(green bar) is from the highlighted row in Table 3.1.

We could see that DGE_{flow} consistently has the best performance. There are two reasons that DGE_{flow} is able to outperform RAW. First, DGE_{flow} employs the multi-hop structural information, which potentially enables the crime to be propagated for more than one hop. Second, DGE_{flow} captures the temporal transition information as well. $LINE$ and MF have worse performance than DGE_{flow}, mainly because embeddings are learned on the independent slotted graph, which does not account for the temporal transition information.

Table 3.3. Average personal income and house price prediction with embedding methods.

Data	Income		House Price	
Features	D+P+G+T		D+P+T	
Method	MAE	MRE	MAE	MRE
RAW	15127	0.250	39.28	0.229
MF	16674	0.2756	39.83	0.233
$LINE$	15534	0.2567	40.438	0.236
DGE_{flow}	-	-	38.95	0.226
$HDGE$	14740	0.2436	-	-

We show the embedding methods comparison of average income and house price prediction in Table 3.3. The income prediction uses the feature combination D+P+G+T, while the house price prediction uses the feature combination D+P+T. Similarly, we observe that the proposed $HDGE$ and DGE_{flow} are able to learn a better relevance scores respectively, and thus improve the RAW method. The other embedding methods MF and $LINE$, however, lead to a worse performance. This verifies that the proposed flow graph design is necessary to account for the relevance among regions.

3.5.2.3 Running Time

We validate the performance gain of applying alias method for random walk sampling on weighted graphs.

In order to validate the efficiency of alias method, we conduct random walk sampling on two flow graphs. The first flow graph is generated at community area level, while the second flow graph is generated at tract level. The tract is a smaller administrative boundary used for the census survey. There are 801 tracts in Chicago, compared to 77 communities areas. The length of random walks for

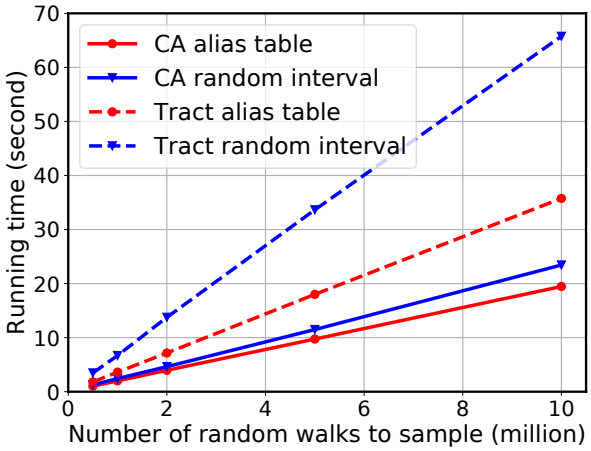


Figure 3.6. The running time of random walk sampling on weighted graphs.

both graph are bounded by 24. The number of sampled random walks ranges from 500k to 10 million.

The running time is shown in Figure 3.6. The compared method is called random interval, which is described in Section 3.4.5.1. It is clear that the alias method consistently runs faster than the random interval method. The alias table method has better performance gain when the number of sampled random walks is large, comparing the solid blue line and solid red line. The reason is that the alias method has a fixed overhead to calculate the alias table for each vertex. Also, the performance gain of alias method on a large graph is bigger. The reason is that alias method reduce the next-vertex sampling complexity from $O(K)$ of the random interval method to $O(1)$, and a larger graph usually has larger K, and thus a larger performance gain.

3.5.3 Interpretations

In this section we give semantic interpretation of the learned dynamic graph embedding. First, we show that $HDGE$ to some degree account for the POI similarity among regions. Next, we use a case study to intuitively explain the semantics captured by $HDGE$.

3.5.3.1 *HDGE* and POI

The POI data reflect different functions of urban areas [28], which is a candidate measure of similarity among regions. Our hypothesis is that to certain degree the *HDGE* accounts for the POI similarity among regions, even though the *HDGE* learning process does not involve any POI data at all.

Due to lack of ground truth, we conduct an unsupervised information retrieval experiment to compare different embedding methods. Each region is used as a query, and the goal is to rank other regions according to their similarities to the query region. The POI similarity ranking is used as the ground truth. The quality of various embedding methods are evaluated with the *nDCG* measures of corresponding rankings.

We use normalized discounted cumulative gain (nDCG) as evaluation measure. Formally, the discounted cumulative gain (DCG) is defined as $DCG@k = \sum_{i=1}^{k} \frac{rel_i}{\log_2(i+1)}$, where the relevance rel_i is derived from POI similarity. The nDCG is the DCG normalized by the idea DCG (iDCG), i.e. $nDCG@k = \frac{DCG@k}{iDCG@k}$, where $iDCG$ is the DCG of the best ranking. Higher $nDCG@k$ value means better quality of the mobility flow embedding similarity.

We conduct this experiment at tract level, and there are 801 tracts in Chicago. We set the embedding dimension as 20 for all methods, and divide one day into 8 3-hour time slots. To make fair comparison, we sample a subset of tracts that all methods are able to learn embeddings, which results in a set of 419 tracts.

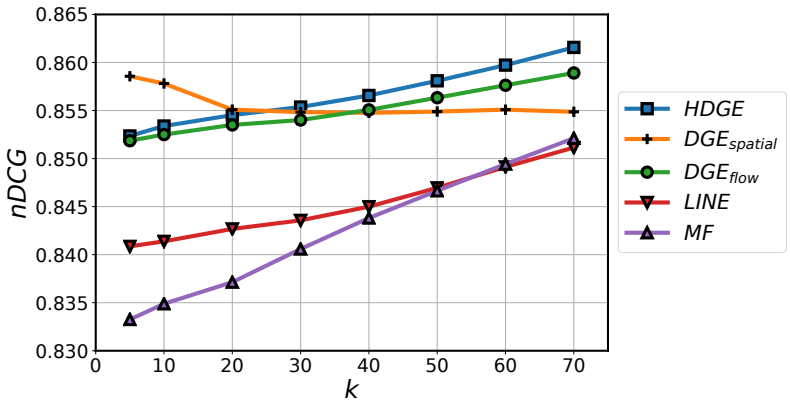
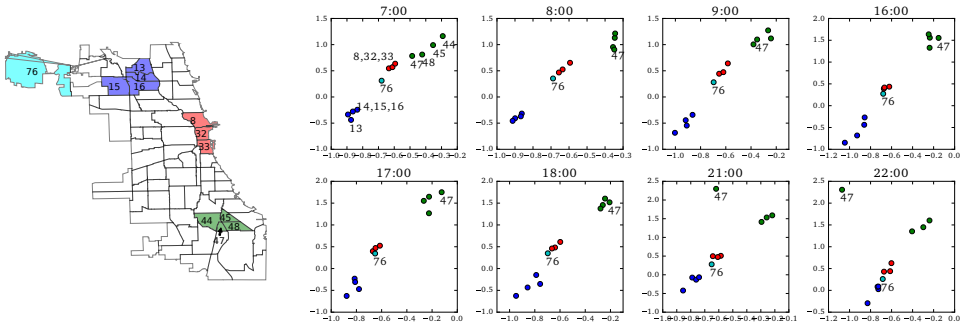


Figure 3.7. The $nDCG@k$ plot for various methods with the pairwise similarity evaluation. k is the number of regions to retrieve.



(a) The positions of selected community areas.

(b) The 2D embedding visualization of selected community areas during different hours.

Figure 3.8. Case study with 2D visualization. We pick 12 communities areas, whose positions in the city are shown in (a). The 2D embeddings from different time are visualized in (b). The 12 communities fall in 4 groups: downtown (red), airport (cyan), residential areas (blue), and residential areas with socio-economic issues (green).

For each embedding method, we report the average $nDCG@k$ across all tracts over all timestamps. The results are shown in Figure 3.7. From the results we made the following several observations.

Overall, the $HDGE$ method significantly outperform other embedding methods, such as MF and $LINE$. This verifies that the design of flow graph accounts for the POI similarity better than the other embedding methods. The reason is that our flow graph not only consider the temporal dynamics, but also draws connection across different timestamps, which is missing in the slotted graph.

It is interesting to notice that when k is small, the $DGE_{spatial}$ gives the best performance, and the performance decreases as the k increases. The reason is that spatially adjacent tracts usually share similar POI distributions. Therefore, given a query tract, a spatial-based method could easily find adjacent tracts as the results for the top 5 other tracts that has the most similar POIs. However, when k is larger than 5, the spatial distance based search does not dominate the results anymore, and thus the performance of $DGE_{spatial}$ decreases.

Although we cannot draw conclusion that $HDGE$ is positively correlate with POI information. This experiment concludes that $HDGE$ design is better than other embedding methods.

3.5.3.2 Case Study

To intuitively demonstrate the semantics of $HDGE$, we learn a 2-dimensional embedding with $HDGE$ method, and visualize 12 hand-picked community areas that represent four different types of areas.

The locations of these 12 community areas are shown in Figure 3.8(a). In Figure 3.8(a), the blue CA 13, 14, 15, and 16 in the north side are densely populated residential areas of the city, where the resident demographics are mostly middle and upper-class. The red CA 8, 32, and 33 locate in downtown, with many commercial, cultural, and financial institutes. In the south of Chicago, the green CA 44, 45, 47 and 48 have different population demographics from the north side. We also plot the Chicago airport, i.e. CA 76, as cyan region. As shown in the map, the Chicago airport locates in the far northwest side of the city, however, it is noteworthy that there are a significant amount of taxi flow commuting between airports and the rest of the city.

In Figure 3.8(b), we visualize the 2-dimensional embedding of these selected regions from different hours. Particularly, we pick three hours in the morning traffic peak, three hours in the afternoon peak, and two hours at night.

As expected, we observe that spatially adjacent community areas are close in the $HDGE$ embedding space. Also, mobility flow helps to identify similar regions beyond spatial adjacency, which explains why the CA 76 is close to downtown area.

An interesting case is observed on CA 47. From the visualization we notice that region 47 has a dramatic change from day to night. During the day time, CA 47 is close to its geographical neighbors, i.e. 44, 45, 48, while at night the embedding of CA 47 is far away from most of the communities. After looking into the taxi trips, we found that there is almost no traffic trip going in or out of CA 47 at night. And the reason behind the extremely low taxi volume is that CA 47 suffers from serious gang violence, so that people are trying to avoid this area at night. In Table 3.4, we show the taxi flow and crime rate of CA 47 compared to its neighbors.

3.6 Related Work

Mobility Data in Urban Problems. Mobility data has been used to solve a wide spectrum of urban problems, such as air quality inference [94], noise pollution

Table 3.4. CA 47 suffers from serious gang-related violence, and thus has much less traffic flows compared to its neighbors. The total number of taxi in/out trips are in 2013. The crime rate is gang-related crime count per 10,000 population in 2013.

CA	In	Out	Crime rate	Crime rank
44	4099	5300	124	7
45	857	1611	112	9
47	**221**	**287**	**185**	**1**
48	1935	2848	72	26

estimation [95], real estate ranking [107], and region function detection [92, 93]. In these existing works, the transition matrix is the most frequently used to represent the mobility flow data. However, the transition matrix ignores the temporal information and the multi-hop transitions. To account for the temporal dynamics, Yuan et al. [28] propose a tensor-based framework to discover regions of different functions, which adds a temporal dimension to the transition matrix. Still, the mobility flow tensor can not capture the multi-hop transitions.

Our method differs from the research mentioned above in how we encode the mobility flow information. We try to encode the dynamic mobility flow into vector representations of regions through a embedding method. The advantage of an embedding method over the transition matrix is that the embedding method preserves the global structural information. More specifically, the transition matrix only preserves the pairwise similarity, while the graph embedding is able to make use of higher order proximity and encode such information into the region representations.

Embedding in Heterogeneous Network. Our method is related to the methods of graph embedding and dimension reduction in general. Some typical methods include multidimensional scaling (MDS) [108], IsoMap [109], Laplacian Eigenmap [110], and graph factorization [111]. These methods find the embedding of a graph by representing the graph as an affinity matrix and then applying matrix factorization. However, the objective of matrix factorization does not necessarily preserve the global network structure, because the matrix factorization only captures the pairwise first-order proximity.

Inspired by the word2vec method from the natural language processing field [96–98], which learns continuous vector representations for words, recent research established an analogy for networks by representing a network as a document [90, 91, 100]. One could sample network by random walk to get sequences of vertices

and learn a continuous representations for each vertex in a low-dimensional space.

When there are multiple types of vertices and edges in the network, the graph embedding learning objective is different. Wang et al. [112] proposed a word embedding method for linked documents, which learns embedding for words, documents, and document labels. Xie et al. [113] apply the heterogeneous embedding technique in a location network to recommend locations.

Our embedding method is applied on a heterogeneous graph as well, but it is still different from most existing works in heterogeneous network embedding. In our problem, we consider a dynamic graph where the relations between the same pair of vertices are changing over time. This new property presents new challenges in embedding learning.

3.7 Conclusion

In this paper, a graph embedding method is proposed to uncover the urban dynamics using mobility flow data. We define a flow graph to incorporate both temporal dynamics and multi-hop transitions. We also define a spatial graph to address the sparsity issue within the flow graph. The dynamic region embeddings are jointly learned from two graphs. With three inference tasks, we demonstrate the effectiveness of our embedding method.

Acknowledgements

The work was supported in part by NSF awards #1544455, #1652525, #1618448, and #1639150. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

Chapter 4
Tackling Spatial Continuity: Task-Specific Region Partition

4.1 Introduction

Recent years have seen more and more cities join the open data initiative. For example, as of December 2016, more than 1600 data sets are available on NYC open data catalog [114]. These public urban data collectively reflect urban dynamics and provide a unique opportunity to fully engage the data-informed city.

Given the growing amount of urban data, a number of data-driven models have been developed to provide insights for urban problems. A frequent approach is to treat each region as a data sample, take the region properties as features, and build a model to learn the correlation between region features and a target variable. One common application lies in the domain of crime prediction. Criminologists are interested in knowing the correlation between demographics and crime [89, 115]. Each region i is taken as a data sample, with X_i as its demographic feature and Y_i as its crime count. A model (e.g., linear regression or negative binomial model) is built to estimate crime count vector Y using the feature matrix X. If some features (e.g., disadvantage index) show a significant correlation with crime count, researchers could relate this empirical result with criminology theory [**?**] and policy makers could further propose corresponding policies to address crime issues.

Existing studies often use pre-defined administrative boundaries (e.g., street block, census tract, community area, or neighborhood) to define a region [28, 89]. An example of the 77 administrative community areas of Chicago is shown in

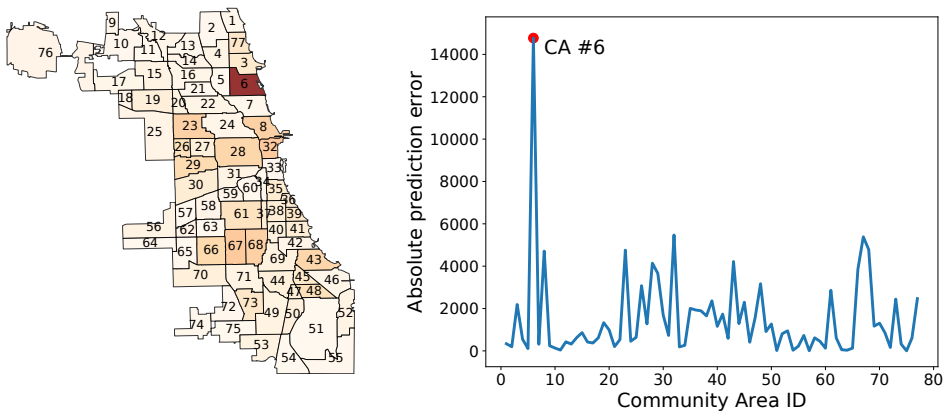
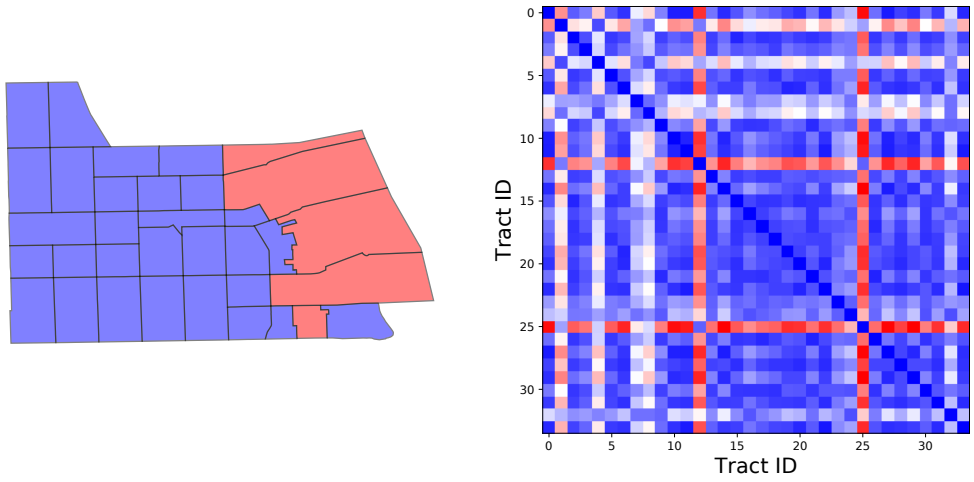


Figure 4.1. Crime prediction error at community level in Chicago. The community area #6 is an outlier with a large error.

Figure 4.1. However, such administrative boundaries might be too rigid and do not reflect the true spatial structure with regards to the targeted urban issues. For example, the definition of regions for crime study might be different from the regions used to understand real estate market. One can alternatively propose to study the problem at the point level or small grid cell level (e.g., 10 meters by 10 meters grids). However, this could lead to data sparsity issues and jeopardize the integrity of the statistical model. Consequently, any inference made from the model would be at risk of bias. These concerns have both theoretical and policy implications. We use the following example to further illustrate the issue of using pre-defined community areas to study crime.

Example 1. *Following previous work [89], we construct a negative binomial model to predict crime count using demographic features by treating each community area as a data sample. Figure 4.1 plots the crime prediction error for each community area of Chicago. Community area #6 (i.e., Lake View area) shows an abnormally high error. In order to explain this outlier, we further investigate the internal structure of this area. Community #6 consists of 34 census tracts as shown in Figure 4.2(a). In Figure 4.2(b), we visualize pair-wise Euclidean distance between tracts based on demographic features. There are five tracts that are different from the other tracts in the demographic feature space, and they are all located on the east side of community #6. These five tracts, when mixed with other tracts in this community, lead to inferior performance of crime count prediction in that area.*



(a) Census tracts of community #6. (b) Tract similarity.

Figure 4.2. Explaining the outlying community area #6 in Figure 4.1. (a) Visualization of the 34 tracts in community #6. (b) The pair-wise similarity of 34 tracts in terms of demographic features. It is clear that five tracts (in red color) on the east side are different from the other blue tracts.

The observation above motivates us to learn a better region definition for crime study. In this paper, we propose a new problem of *task-specific region partitioning*. Given spatial variables and a selected model (e.g., linear regression), we aim to partition the city into regions such that the model trained by taking regions as data samples achieves the optimal results.

To the best of our knowledge, task-specific region partitioning is a new problem that has not been studied before. While there exist many methods for spatial clustering [116], most of them group the locations based on the similarity of their spatial properties and do not have a target variable to predict. Another similar problem is to partition the locations into k regions and fit a model for each region. Such a problem definition has a different purpose with the goal of showing that the correlations between features and target vary over the space (e.g., in some area, disadvantage index correlates with crime; while in some other areas, it does not). In our problem definition, we aim to fit one model for the whole city hoping to get a generalized interpretation (e.g., disadvantage index significantly correlates with crime count in Chicago). Such a problem definition is a frequently adopted form in the criminology literature [**?**].

Task-specific city region partitioning is a challenging problem. The key challenge lies in that the region properties (both features and target variable) and the model coefficients change simultaneously when we change the region partition. We prove that this is an NP-hard problem. In our proposed solution, we employ the Markov Chain Monte Carlo (MCMC) method. We start from a pre-defined region partition (e.g., community areas), and generate a new partition sample by flipping the membership of a smaller area (e.g., a tract). Two variants of MCMC methods are proposed to solve this problem. First, a naive MCMC method generates the next partition sample by randomly flipping a tract. Second, a heuristic-based MCMC method generates the next sample by flipping one tract randomly selected from the community areas with the highest error. Finally, we employ reinforcement learning to automatically learn how to generate the next sample that is more likely to improve the prediction performance.

We evaluate our method on two real datasets, i.e. crime count and real estimate price. The learned region partitions are shown to consistently outperform the administrative boundaries and spatial clustering method. For example, our methods, on average, outperform the administrative boundary by 56% in a crime prediction task. We also observe that the heuristic-based MCMC converges faster than the naive MCMC, while the reinforcement learning uses the least iterations to converge.

To summarize, the key contributions of this paper are:

- We propose a novel problem on task-specific region partitioning. This problem is motivated by real-world urban studies, including our own previous work on crime prediction [89].
- We prove the problem is NP-hard and we study different MCMC and reinforcement learning sampling techniques to solve the problem.
- We validate our method through extensive experiments on two real datasets.

The rest of this paper is organized as follows. The formal definition of our problem is given in Section 4.3, and our method is described in Section 4.4. Section 4.5 shows the evaluations on two different tasks. Section 4.2 summarizes related work. Finally, we conclude in Section 4.6.

4.2 Related Work

Urban Data Heterogeneity. Various urban data exhibit high degrees of correlation. As we collect more types of new urban data, we are able to solve a wide spectrum of urban problems. For example, a real-time air quality inference system is proposed in [94], which uses not only historical air quality data, but also traffic flows, structure of roads, and POIs. Zheng et. al. [95] diagnose New York City noise pollution with complaint records, road networks, and human check-ins. Real estate values are predictable given online user reviews [107] and offline human mobility data [117]. Wang et. al. [89, 115] improve crime prediction accuracy by combining POI data and taxi flow data.

These existing works focus on mining the subtle correlations across different domains of data. We generalize the urban problems above as a learning task ,f, which maps some urban features to a target variable of interest. In this paper, we use crime prediction and average house price prediction as two examples. We study how to define the domain of urban problem, f, because only when f is defined over a proper unit of study (e.g. community areas), the learned correlation is consistent and significant.

Traditional Region Partition Methods. Our problem falls into the region segmentation category. There are four main types of region partition methods that are widely used in the urban computing literature. First, a fixed sized grid is the most straightforward partition for travel time prediction [118], interpret traffic dynamics [119], and air quality inference [94]. Second, existing administrative boundaries are also used for crime prediction [89]. Third, clustering of point-wise urban data to get regions. For example, Li et. al. [120] study the bike-sharing system and propose to estimate the supply/demand of bikes in a station cluster. Finally, other partitions are specifically designed for special needs. For example, Yuan et. al. [28] employ the major road networks to partition a city into regions and learn the function of each region. Xu et. al. [121] partition a city by cellular tower coverages to study the mobile traffic pattern in urban environment. Zheng et. al. [122] use fan-shaped partitions to predict fine-grained air quality, because wind direction is an important indicator.

While various partition methods are used in existing urban problems, none of

the partition methods explicitly take the learning task f into consideration. It is worthy mentioning that most existing partition methods are purely based on cartographic information, and do not make use of the urban data properties. In this paper, we try to partition the city with an explicit objective.

Discrete Optimizations. The objective of our problem is a discrete optimization problem and is easy to derive. However, since the problem is NP-complete, it is challenging to efficiently find an optimal solution. MCMC sampling has been shown to be effective in optimizing discrete structures [123]. We follow this line of work and propose to use MCMC sampling to search for the optimal partition. During the MCMC sampling process, a lot of partitions are sampled but do not achieve better prediction results. To solve this issue, we follow the learning to optimize technique [124], and propose to employ the reinforcement learning framework to learn where to sample the next partition that is most likely to improve the learning task, f.

4.3 Region Partition Problem

In this section, we first present the formal definition of our method. Then we prove that this partition problem is NP-hard.

The input of our problem is a set of minimum spatial units. Without loss of generality, we use tract as the unit of study in this paper, and the set of tracts is denoted as $\mathcal{T} = \{t_1, t_2, \cdots, t_n\}$. Within each tract t_i, the following information are available $\langle p_i, y_i, x_i \rangle$, where p_i is a sequence of GPS coordinates representing tract exterior boundary, y_i is the target variable of interest, such as crime count and average house price, and $x_i \in \mathcal{R}^d$ is a d-dimension contextual feature vector. Note that each tract is a polygon with one connected component.

We define community area as the proper unit to study the correlation between y and x.

Definition 5 (Community area)**.** *Each community area consists of several adjacent tracts, denoted as* $Z_j = \{t_1^j, t_2^j, \cdots\}$*. We derive* $\langle P_j, Y_j, X_j \rangle$ *for each community area* Z_j *by aggregating* $\{\langle p_i, y_i, x_i \rangle | t_i \in Z_j\}$*, where* P_j *is the exterior boundary,* Y_j *is the crime count, and* X_j *is the contextual features of* Z_j*.*

Note that there are various approaches to aggregate tract-level features x_i into community area features X_j. For example we can use element-wise summation, min, or max on categorical count features. We can also calculate entropy to derive a diversity feature as X_j.

With the definition of community area, we define the partition of a city.

Definition 6 (Partition)**.** *A partition over $\mathcal{T}$ is denoted as $\mathcal{Z} = \{Z_1, Z_2, \cdots, Z_m\}$, satisfying the following four conditions*

1. *(subset) $\forall j$, $Z_j \subset \mathcal{T}$;*
2. *(non-overlapping) $\forall p, q$, $Z_p \cap Z_q = \emptyset$;*
3. *(completeness) $\bigcup_{j=1}^{m} Z_j = \mathcal{T}$;*
4. *(spatial-continuity) $\forall j$, P_j defines a polygon with exact one connected component.*

A task is defined on a given partition of the city, where we aim to learn the correlation between X and Y at the community area level.

Definition 7 (Task)**.** *Given a partition $\mathcal{Z}$, a task is to learn a linear function f, such that $Y_j = f(X_j)$, where X_j and Y_j are target variable and contextual features of community area Z_j.*

Our ultimate goal is to have a more accurate prediction. The task-specific region partition problem is defined as follows.

Definition 8 (Task-specific region partition)**.** *Given a set of tracts $\mathcal{T}$ and a task f, find a partition $\mathcal{Z}$ with m components, such that the task error is minimized. Formally, we have*

$$\arg\min_{\mathcal{Z},f} \sum_{j=1}^{m} \Big(||Y_j - f(X_j)||_2 + G(Z_j) \Big), \tag{4.1}$$

where $G(\mathcal{Z}) = \sum_{j=1}^{m} G(Z_j)$ is a constraint function on partition.

The constraint function G ensures the partition $\mathcal{Z}$ has desirable property, such as small variance in community populations or balanced size in terms of community

area.

NP-Hardness. The problem in Definition 8 is a combinatorial optimization problem, where the set of instances is $\mathcal{T}$, the feasible solution is $\mathcal{Z}$, and the quality measure of solution is $\mathcal{F}(\mathcal{Z}, f) = \sum_{j=1}^{m} \mathcal{F}(Z_i, f)$. The decision version of the problem is to find a partition $\mathcal{Z}$, such that $\mathcal{F}(\mathcal{Z}, f) \leq \epsilon$, where ϵ is a constant. In this section, we prove such decision problem is NP-complete, and therefore the optimization problem in Definition 8 is NP-hard.

In the NP-completeness proof, we approximate the decision problem above with an easier problem. The reason is that both X_i, Y_i, and f are dynamically changing according to $\mathcal{Z}$, which complicates the original problem. First, we replace the jointly learned optimal f with a fixed f_0. Since f is optimal to minimize Equation 4.1 while f_0 is not, we have $\mathcal{F}(\mathcal{Z}, f) < \mathcal{F}(\mathcal{Z}, f_0)$. Second, we use $\sup\{\mathcal{F}(Z_j, f_0)\}$, where sup calculates supremum of a set, to approximate $\mathcal{F}(\mathcal{Z}, f_0)$. Since there are finite number of tracts within each community Z_j, and the task function f_0 can be solved in polynomial time, therefore the upper bound of $\mathcal{F}(Z_j, f_0)$ exists and is finite. Combine two approximations above, we have $\mathcal{F}(\mathcal{Z}, f) < m \cdot \sup\{\mathcal{F}(Z_j, f_0)\}$. The approximated decision problem is to find a partition $\mathcal{Z}$, such that $m \cdot \sup\{\mathcal{F}(Z_j, f_0)\} \leq \epsilon_0$.

Now we prove the approximated decision problem is NP-complete. First, such approximated decision problem is NP, because given a partition $\mathcal{Z}_0$, we are able to validate $m \cdot \sup\{\mathcal{F}(Z_j, f_0)\} \leq \epsilon_0$ in polynomial time. Next, we prove the NP-hardness of this problem by reducing the (k, v)-balanced partitioning problem to the approximated decision problem. The (k, v)-balanced partition problem [125] is a proved NP-complete problem, which partition graph into k disjoint components of size at most $v\frac{n}{k}$, while the capacity of edge cut is less than ϵ_0. We construct the adjacency graph of all tracts $\mathcal{T}$, with weight $\sup\{\mathcal{F}(Z_j, f_0)\}$ on each edge. A solution to (m, v)-balanced partition problem on such adjacency graph is a solution to the approximated decision problem. The balanced partition problem achieve $k \cdot \sup\{\mathcal{F}(Z_j, f_0)\} \leq \epsilon_0$, where k is the number of edges to cut the graph. It is clear that $k > m$, and therefore we find a partition satisfying $m \cdot \sup\{\mathcal{F}(Z_j, f_0)\} \leq \epsilon_0$.

The original problem in Definition 8 is NP-hard. Therefore, it is difficult to efficiently search for the optimal partition. In this paper, we use Markov Chain Monte Carlo sampling strategy to search for local optimal solutions.

4.4 methods

In this section, we use the stochastic Markov Chain Monte Carlo (MCMC) method to automatically learn the partition $\mathcal{Z}$. We propose two variations of MCMC sampling method with different sample proposal strategy. And finally, we use reinforcement learning to make the best of historical samples and automatically learn the sample strategy.

4.4.1 Markov Chain Monte Carlo

There are two parameters, $\mathcal{Z}$ and f, to learn in Equation 4.1. However, given a fixed partition $\mathcal{Z}$, the optimal task function f can be easily learned. The challenge lies in searching through the partition space. Toward this goal, we adopt the MCMC method, or more specifically the Metropolis-Hastings algorithm to optimize $\mathcal{Z}$.

Markov Chain Monte Carlo [126] is a stochastic algorithm for obtaining a sequence of random samples from a distribution for which direct sampling is difficult. A key property of the algorithm is that it constructs a Markov chain that will ultimately converge to p through stochastic sampling [127]. In our case, the state space is all possible partitions $\mathcal{Z}$, and the distribution $p(\mathcal{Z})$ defines the probability that $\mathcal{Z}$ is optimal to Problem 8. Clearly, it is difficult to calculate $p(\mathcal{Z})$, however the quality measure $\mathcal{F}(\mathcal{Z})$ is proportional to $p(\mathcal{Z})$. Namely, a partition $\mathcal{Z}$ with lower $\mathcal{F}(\mathcal{Z})$ value is more likely to be optimal.

In addition to the quality function $\mathcal{F}$, MCMC employs a proposal function $q(\mathcal{Z}'|\mathcal{Z})$, which defines the transition probability from state $\mathcal{Z}$ to $\mathcal{Z}'$. The Markov chain moves toward $\mathcal{Z}'$ with acceptance probability γ, defined as

$$\gamma = \min\left[1, \frac{p(\mathcal{Z}')q(\mathcal{Z}|\mathcal{Z}')}{p(\mathcal{Z})q(\mathcal{Z}'|\mathcal{Z})}\right]. \tag{4.2}$$

$p(\mathcal{Z})$ is the Boltzmann distribution, defined by

$$p(\mathcal{Z}) = \frac{e^{-\mathcal{F}(\mathcal{Z})/T}}{P}, \tag{4.3}$$

where P is the normalization constant, and T is the temperature parameter. We do not explicitly compute P, because they cancel out in Equation (4.2).

The MCMC algorithm is shown in Algorithm 1. First, we initialize the partition

Algorithm 1 MCMC method to search $\mathcal{Z}$.

$\mathcal{Z} \leftarrow \mathcal{Z}_0$
while $\mathcal{F}(\mathcal{Z}) \geq \epsilon$ **do**
    Sample $u \leftarrow \mathcal{U}_{[0,1]}$
    Sample $\mathcal{Z}' \leftarrow q(\mathcal{Z}'|\mathcal{Z})$
    $\gamma = \min\left[1, \frac{p(\mathcal{Z}')q(\mathcal{Z}|\mathcal{Z}')}{p(\mathcal{Z})q(\mathcal{Z}'|\mathcal{Z})}\right]$
    if $u < \gamma$ **then**
        $\mathcal{Z} \leftarrow \mathcal{Z}'$
    end if
end while

with the existing administrative boundary, denoted as $\mathcal{Z}_0$. Within each step, we draw $u \in [0, 1]$ from uniform distribution $\mathcal{U}_{[0,1]}$, and draw the next partition $\mathcal{Z}'$. The acceptance probability γ is calculated according to Equation 4.2. If u is smaller than the acceptance probability γ, then we accept the new partition $\mathcal{Z}'$. We repeat the process above, until $\mathcal{F}(\mathcal{Z})$ converges.

4.4.2 MCMC with Naive Proposal Distribution

The proposal function q is very flexible, if not limitless. In practice, the exact form of q generally affects the efficiency of the algorithm, or how quickly it will converge. We begin by establishing a baseline MCMC approach to the region partition problem by outlining the simplest q we can devise.

We generate a new partition by randomly selecting one tract $t_i \in \mathcal{T}$ that is on the boundary of some community area Z_j, and then flip t_i to the adjacent community area. Such a naive proposal function q contains the following steps.

- Maintain a set of tracts that are on partition boundary, denoted as $\mathcal{T}_b$.
- Uniformly draw t_i from $\mathcal{T}_b$. The current community assignment for t_i is Z_j. The probability of selecting t_i is $1/|\mathcal{T}_b|$.
- Uniformly draw Z_p from the set of adjacent community areas $Adjacent(t_i)$ of t_i. The probability of selecting Z_p is $1/|Adjacent(t_i)$.
- Verify the four conditions in Definition 6 are satisfied. If not, restart.
- Assign the t_i to community Z_p. Update boundary set $\mathcal{T}_b$.

The naive proposal function q is symmetric, because $q(\mathcal{Z}'|\mathcal{Z}) = q(\mathcal{Z}|\mathcal{Z}') = \frac{1}{|\mathcal{T}_b| \cdot |Adjacent(t_i|}$. Therefore, the acceptance probability $\gamma = \min[1, \frac{p(\mathcal{Z}')}{p(\mathcal{Z})}]$.

4.4.3 Guided MCMC with Softmax Proposal Distribution

The naive proposal MCMC method is simple, but not efficient. The reason is that we are randomly trying different partitions. We now propose an MCMC approach with a more intelligent proposal strategy. This method follows a greedy intuition that we should adjust the community area with the highest prediction error to improve current partition.

Given this intuition, we heuristically design our guided MCMC method to sample a community area with large error first. To achieve this, we apply the softmax function over the prediction errors on community areas to derive the sample probability of each community area. The softmax proposal contains the following steps.

- Maintain a set of tracts that are on partition boundary, denoted as $\mathcal{T}_b$.
- Draw Z_j from $\mathcal{Z}$ proportional to the prediction of error using softmax function, $p(Z_j) = \frac{\exp(||Y_j - f(X_j)||_2)}{\sum_{k=1}^{m} \exp(||Y_k - f(X_k)||_2)}$.
- Uniformly draw a tract t_i from $Z_j \cap \mathcal{T}_b$.
- Uniformly draw Z_p from the set of adjacent community areas $Adjacent(t_i)$ of t_i. The probability of selecting Z_p is $1/|Adjacent(t_i)$.
- Verify the four conditions in Definition 6 are satisfied. If not, restart.
- Assign the t_i to community Z_p. Update boundary set $\mathcal{T}_b$.

Note that under the softmax proposal approach, the proposal function q is not symmetric. Therefore, we have to explicitly calculate the values for q function and plug into Equation 4.2.

4.4.4 Reinforcement Learning

There are two main drawbacks of the MCMC method. First, when drawing a new sample, the MCMC methods do not account for any information from previous

samples. For example, the naive proposal MCMC rejected early generations of samples, which do not contribute any information to future generations of samples. Intuitively, if we repeatedly observe that flipping some tracts give us lower gain, then we should lower the probability that such tracts are sampled again in the future. Second, the Markov chain-based stochastic search strategy is more likely to get stuck on a local optima. The reason is that the nature of MCMC sampling follows a depth first search in the huge search space. It is very likely that another chain exists that leads to a better local optima, but the algorithm is not able to achieve this better outcome.

To address the issues of the MCMC method, we further propose a reinforcement learning (RL) [128] scheme for generating new samples. RL differs from standard supervised learning in that the correct input/output pairs are never presented. Instead, RL is concerned with how agents ought to take actions in an environment so as to maximize cumulative gain.

In what follows, we map the RL components to our problem. The set of tracts consists of the environment, and their community area assignment is the state. An action is to re-assign some tract t_i from Z_j to Z_p, denoted as tuple $\langle t_i, Z_p \rangle$. The immediate reward of such transition is defined by $\Delta\mathcal{F} = e^{-\mathcal{F}(\mathcal{Z}')} - e^{-\mathcal{F}(\mathcal{Z})}$, where the exponential function coverts the loss into gain. We define the cumulative gain Q as a function of the current state $\mathcal{Z}$ and action $\langle t^k, Z^k \rangle$ at step k, and Q satisfies the following condition

$$Q(\mathcal{Z}, \langle t^k, Z^k \rangle) = \Delta\mathcal{F} + \delta \cdot \sum_{a \in \{\langle t^{k+1}, Z^{k+1} \rangle\}} Q(\mathcal{Z}', a), \tag{4.4}$$

where δ is the discount factor on future reward and $\{\langle t^{k+1}, Z^{k+1} \rangle\}$ is the set of all possible actions given $\mathcal{Z}'$. Given Q function, at each state $\mathcal{Z}$, we are able to find the best action with a the highest cumulative gain through

$$\arg\max_{\langle t, Z \rangle} Q(\mathcal{Z}, \langle t, Z \rangle). \tag{4.5}$$

In our problem, a reinforcement learning scheme faces the following three challenges. We devise specific approximations to address these challenges.

Huge state space. The state space is exponential to the number of tracts.

As a consequence, we cannot track the exact reward for each state and actions. Instead, we rely on Deep Q-learning [129], where a deep neural network is learned to approximate the Q function. Our neural network structure includes an embedding layer to encode the partition $\mathcal{Z}$ and action tuple, two dense fully connected dense layers, and finally the output layer predicts the sigmoid of Q.

Large and dynamic action space. The number of possible actions is linear to the number tracts. Also, for different partition states, the tracts on the boundaries are different, and thus the action set is also different. This property makes it difficult to calculate the summation of future Q values in Equation (4.4), and to find the maximum over all actions in Equation (4.5). In this paper, we set the discount factor $\delta = 0$ to ease the calculation of Q. When searching for the best action with Equation (4.5), we sample a subset of $m = 32$ actions, and find the best action within such subset.

Training overhead is high. To train the Deep Q-learning model, we do a sequence of random action to generate a batch of training data. Such training overhead is significantly higher than the MCMC method. To improve the efficiency, we save our Deep Q model across different tasks and different rounds. The neural network is only updated when found action cannot improve the cumulative reward.

4.5 Experiment

In this section, we first describe the datasets and experimental settings. Then we quantitatively evaluate our method on two separate prediction tasks. Finally, we present two case studies to intuitively explain the strength of our methods.

4.5.1 Experiment Setting

4.5.1.1 Data description

The fundamental geographic unit of study in this paper is a tract, which is a small area established by the U.S. Census Bureau for analyzing populations. We use the tract as the unit of study, because it offers the finest granularity for which demographic data is recorded. The following data are used in this paper, and a summary of the data property is given in Table 4.1.

Table 4.1. Data set property

Data set	Granularity	#Sample	#Field	Year
Demographics	tract	801	110	2010
Map	tract	801	-	2010
Crime	point-wise	719,461	16	2010-2011
House price	point-wise	44,447	14	2015-2017

Demographic Data. Socioeconomic and demographic features of neighborhoods have been widely used to predict crime [118]. We collect the demographic data of 801 tracts in Chicago from 2010 US census survey [130]. The demographics data provide the raw count of households over 100 different categories, include ethnicity, income, education, etc.

Map Data. The geographic boundary information for 801 tracts are available through the U.S. census survey [130] as well. Each boundary is defined by a polygon with a sequence of GPS coordinates.

Crime Data. The Chicago Data Portal [105] provides the detailed records of over five million incidents of crime from 2011 - 2017. Our study focuses on year 2010 and 2011 only. For each crime incident, the date, location, and incident type are reported.

House Price Data. House price data in the Chicago metropolitan area are obtained from Zillow, a popular real estate website [106]. We collect the last sale price, floor size, latitude, and longitude information for over 44,000 properties that were sold between January 2015 and December 2017.

4.5.1.2 Prediction Tasks

We employ a negative binomial regression model [89] as our prediction task, namely

$$\mathbf{E}(Y) = \exp(\alpha X),$$

where $\mathbf{E}$ calculates the expectation. The link function used in the regression is a negative binomial distribution. The advantages of negative binomial regression are two-fold. First, a negative binomial model is suitable for non-negative value prediction. Second, compared to Poisson regression, negative binomial regression solves the over-dispersion problem by allowing the variance to be larger than the mean.

The demographics features at the community level are used as contextual features X_i. Note that it is not feasible to directly use the raw count in the demographic data as X. We pre-process the raw counts into mutually independent features, such as total population, percentage of household in each income category, diversity of ethnicity, etc.

Two prediction tasks are used to evaluate our region partition methods in the experiments. Both tasks use the same processed demographic features as X. The first task is crime count prediction, where we aggregate the total number of crime in a community as Y. The crime count in year 2010 is used as training data, and the crime count in 2011 is used as testing data. The second task is house price prediction, where we use the average price per square foot in a community as Y. The houses that are sold before August 1st, 2016 are used as training data, while the rest are used as testing. The split point makes the training and testing data have a roughly equal number of samples.

4.5.1.3 Compared Methods

The existing administrative boundary is a clear baseline partition. Since our region partition problem is similar to clustering problem, we employ various conventional clustering methods to derive different clustering partitions as alternative baselines. More specifically, we compare the following region partition methods.

- **(`Admin`)** Administrative boundary uses the existing administrative boundary defined by US Bureau of Census [130]. A visual depiction of this partition can be seen in Figure 4.1.
- **(`Agglomerative`)** Agglomerative clustering performs a hierarchical clustering using a bottom up approach. The ward linkage function is used, along with a tract adjacency graph as input to guarantee spatial continuity.
- **(`K-Means`)** K-means clustering separates tracts into n groups of equal variance.
- **(`Spectral`)** Spectral clustering does a low-dimensional embedding of the affinity matrix between samples first, and then applies the K-means method in the lower dimensional space. Note that `Spectral` also takes the tract adjacency graph as the affinity graph.

(a) `Agglomerative` (b) `K-Means`

(c) `Spectral` (d) `DQN`

Figure 4.3. (a - c) The clustering results for three clustering baselines. (d) The learned partition from `DQN` method for crime prediction task.

- **(`Naive`)** MCMC with naive proposal. The first variant of our proposed MCMC method using a straightforward uniform proposal.
- **(`Softmax`)** MCMC with softmax proposal is another variant of our MCMC method, which uses strong heuristics.
- **(`DQN`)** Q-learning is our proposed reinforcement learning method to search for optimal partition.

For various clustering methods, we set the number of clusters as $m = 77$, which equals the number of community areas in `Admin`. Note that if we run clustering methods multiple times, they usually produce the exact same clustering results. The MCMC and `DQN` methods, on the other hand, are all stochastic processes and do not converge to the same partition. Therefore, we run 100 rounds of our proposed methods and report the average measure.

4.5.1.4 Evaluation Metrics

Given a partition $\mathcal{Z}$, we evaluate the quality of this partition on the testing data set. Mean absolute error (MAE) is used to measure the performance of prediction tasks, i.e.

$$MAE = \frac{\sum_{j=1}^{m} |Y_j - \hat{Y}_j|}{m}, \tag{4.6}$$

where $\hat{Y}_i$ is the leave-one-out prediction error of community Z_j. Namely, we train a model on the rest of the communities $\mathcal{Z} \setminus Z_j$. Then, $\hat{Y}_j$ is the estimated target value for Z_j from the trained model.

4.5.2 Quantitative Evaluations

4.5.2.1 Effectiveness Study

In Table 4.2, we report the evaluation results of the various partition methods. The partitions results from different baselines are visualized in Figure 4.3(a-c). Since our methods are run for 100 rounds, we report both the MAE and its standard deviation in the table. The final partition of `DQN` is visualized in Figure 4.3(d). Overall, we have the following three observations.

Clustering methods overall perform poorly. This is likely due to the fact that the clustering methods do not consider the task information. More specifically, `Agglomerative` results in the highest prediction errors for both crime prediction and house price prediction task. The reason is that `Agglomerative` method utilizes tract connectivity as a hard constraint. As a result, the generated communities have a large variance in their sizes, as shown in Figure 4.3(a). `K-Means` gives worse result than that of `Admin` as well. The generated partition of `K-Means` seems to consist of more than m communities in Figure 4.3(b) because `K-Means` does not incorporate spatial continuity constraint. Consequently, one community can

Table 4.2. Prediction MAEs of various partition methods. Our proposed methods are run for 100 rounds. The MAE and its variance are reported.

Method	MAE	
	Crime	House price
`Admin`	1715.91	31.29
`Agglomerative`	72201.00	50.34
`K-means`	2887.83	32.40
`Spectral`	1440.57	29.66
`Naive`	1073.42(81.93)	25.73(2.76)
`Softmax`	1041.68(76.75)	27.13(2.98)
`DQN`	**746.13**(154.19)	**25.16**(1.30)

consist of several disconnected components. `Spectral` methods generates the best results in both tasks among these baselines because `Spectral` accounts for the affinity of tracts and generates communities with similar sizes. However, it is worth mentioning that `Spectral` method cannot guarantee the spatial continuity of generated communities. From Figure 4.3(c), we can also see that `Spectral` shows similar partition as the original administrative boundary, shown in Figure 4.1(a). That is also why it achieves similar prediction accuracy as `Admin`.

The proposed MCMC method outperforms the baselines. Both variants of MCMC methods significantly outperform `Admin` and `Spectral` baselines. Such observations validate the effectiveness of the MCMC strategy in searching for optimal solutions. However, it is not conclusive to say `Softmax` is better than `Naive`, because while `Softmax` has better performance on crime prediction task, `Naive` has better performance in house price prediction. The reason could be that the heuristics used in `Softmax` is not universally applicable. The heuristic assumes that working on a community with the highest error will lead to optimal solution. When this heuristic is wrong, it aggressively reduces the search space, excluding where there are better local optimal partitions.

`DQN` method performs the best among all. It is clear that `DQN` finds a better local optimal solution than that of MCMC methods. On the crime prediction task, the average MAE is 746.13, which represents a 56% improvement over `Admin` baseline. On the house price prediction task, `DQN` consistently gives the best performance. The reason is that `DQN` explores over a subset of actions and picks the best one at each step, compared to the MCMC method, which searches the partition space in a depth first search fashion. Comparing the final partition of `DQN` and `Spectral`, we

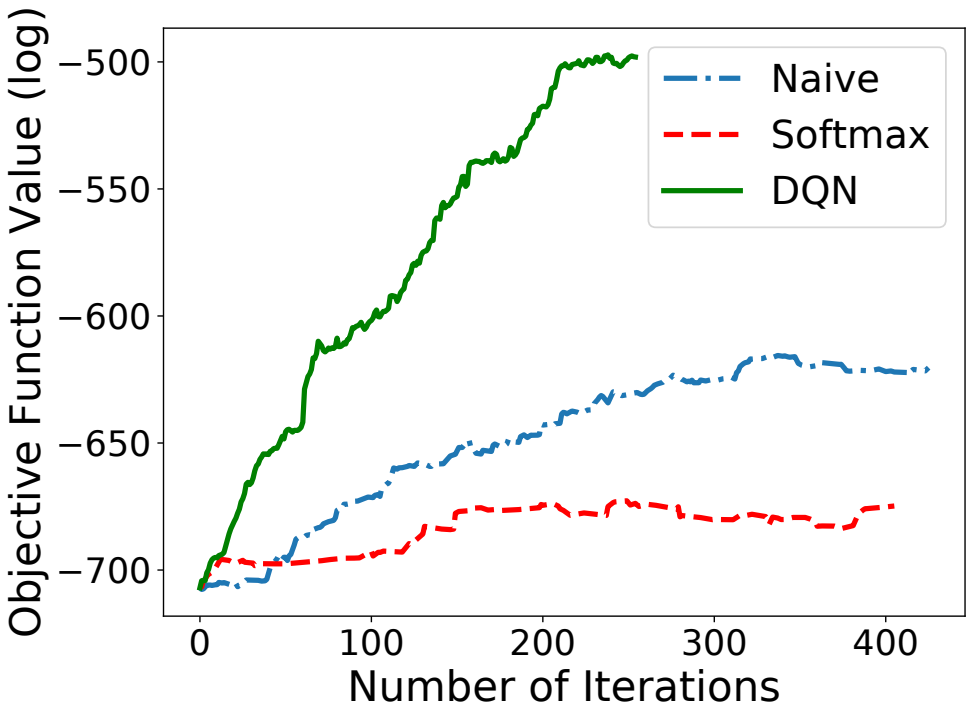


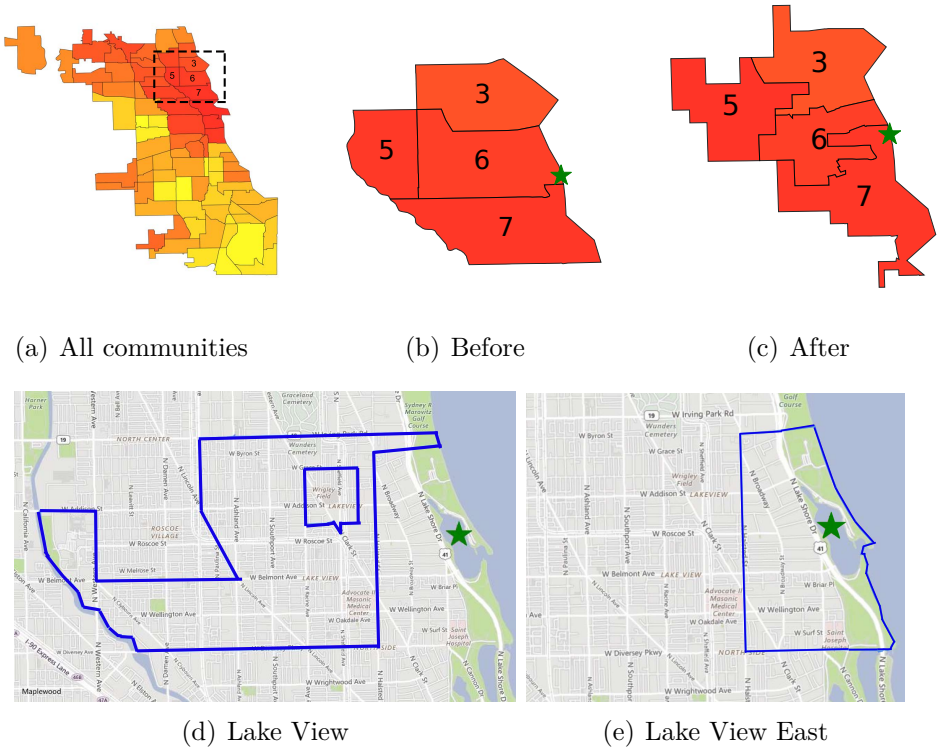
Figure 4.4. Convergence plots for proposed methods on house price prediction task.

will notice that `Spectral` partition is more similar to the original administrative boundary in Figure 4.1. As a consequence, `Spectral` has similar MAE to `Admin`, but is much worse than `DQN`.

4.5.2.2 Convergence Study

We conclude the effectiveness study with a brief comparison of the convergence of three proposed methods. In our method, we track the standard deviation of the $\mathcal{F}$ values from last 50 iterations. When such standard deviation is less than a pre-defined threshold, we stop. In Figure 4.4 we visualize log of quality measure, i.e. $-\mathcal{F}(\mathcal{Z})$, against the number of iterations for three proposed methods on the house price task.

We observe that `DQN` finishes in a less than 300 iterations, while `Naive` and `Softmax` both take more than 400 iterations. Clearly, we observe that `DQN` converge to a better optimal solution with higher training gain. Comparing with `Naive`, `Softmax` converges faster at the beginning, because of the strong heuristics behind. However, `Naive` eventually finds a better local optimal solution than that of



(a) All communities (b) Before (c) After

(d) Lake View (e) Lake View East

Figure 4.5. House price prediction case near Belmont Harbor, denoted by green star. (a) Average house price distribution in Chicago under `Admin` partition. Dotted rectangle denotes region of interest. (b) Region of interest under `Admin` partition. (c) Region of interest under `DQN` partition. (d - e) Region of interest according to Zillow. Note that the Belmont Harbor area is split into a separate community, named Lake View East.

`Softmax`, because the heuristics eliminates search space too aggressively, such that the algorithm could not explore search space with better local optimal partitions.

4.5.3 Case Studies

In this section, we present two interesting case studies of the region partitions learned from `DQN` method.

House price prediction case study. We present a case study near community #6, Lake View, in the house price prediction task, as shown in Figure 4.5. This case shows that `DQN` partition is actually superior to the original administrative boundary, because `DQN` partition matches better with expert domain knowledge.

We visualize a heat map of house price (per square foot) for the whole city of Chicago in Figure 4.5(a). Warmer colors (red) denote higher prices, while cooler

colors (yellow) indicate lower prices. The dotted rectangle in the figure marks the region of interest for our discussion, which is community #6. A zoom-in view of this area using the administrative boundary is shown in Figure 4.5(b). Notice that all nearby areas have relatively high average house price.

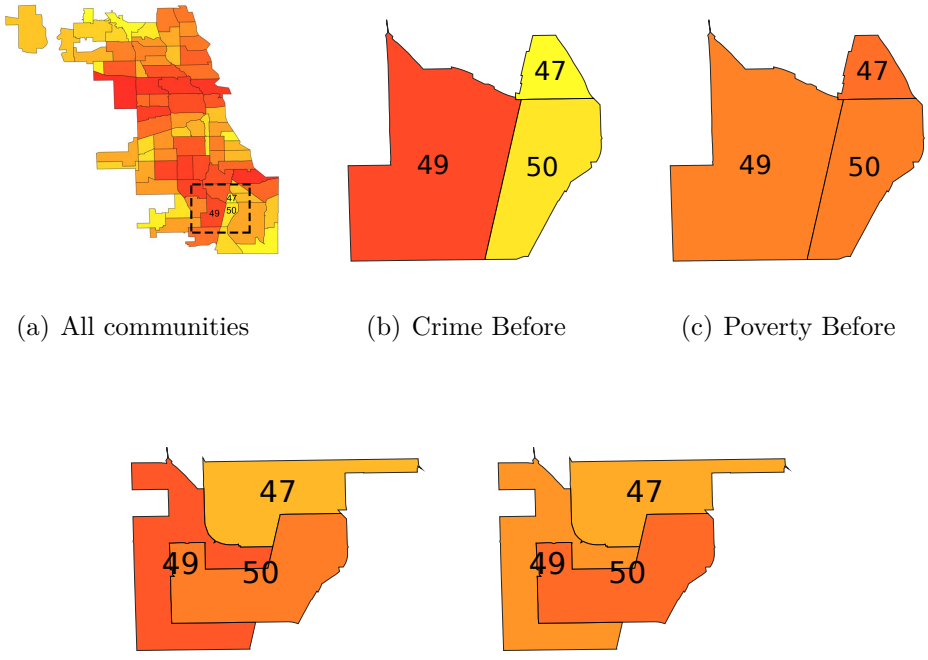
Recall from the Figure 4.2 that east side of community #6 is different from the rest area of community #6. We further find the following evidence to support the fact that the east side of community #6 is different from the rest area. The coastal region of the original community #6 contains Belmont Harbor, one of Chicago's largest boating areas. Notice that `DQN` method divides community area #6 and groups the coastal tracts surrounding the Belmont Harbor with other coastal areas farther south in community #7. These areas contain other leisure destinations such as the Lincoln Park Zoo, and numerous beach areas. This semantic argument is bolstered by an independent source: Zillow's own self-defined regions. Figure 4.5(d) shows that the Lake View area does not contain Belmont Harbor. Meanwhile, Zillow assigns Belmont Harbor as a separate region called Lake View East, as shown in Figure 4.5(e).

Surprisingly, `DQN` partition is similar to Zillow's self-defined region in that they both exclude the Belmont Harbor area from the original community #6 (Lake View neighborhood), as shown in Figure 4.5(c). While the two regions are not exactly the same, it is interesting to note that they both remove the Belmont Harbor area from the original community.

Crime prediction case study. For crime study, we give the intuition why our partition gives lower prediction errors in the corresponding task. The case is shown in Figure 4.6.

For the purpose of simplicity, we choose a single feature, that reasonably correlates with our variable of interest, namely crime count. The single feature we choose is the poverty index, which is calculated by the percentage of households in a given community whose combined income is less than or equal to \$30, 000. Figure 4.6(a) shows a heat map of crime count by community, using the original administrative boundary. Warmer areas correspond to higher crime counts. We focus on communities #47, #49, and #50, as marked by the dotted rectangle in Figure 4.6(a).

A zoom-in plot of the crime count by administrative boundary is shown in Figure 4.6(b). Additionally, we construct a similar plot of the poverty index of these



(a) All communities (b) Crime Before (c) Poverty Before

(d) Crime After (e) Poverty After

Figure 4.6. Crime prediction case near Community #47. (a) Crime count distribution in Chicago under `Admin` partition. Dotted rectangle denotes the region of interest. (b) Crime count in region of interest under `Admin` partition. (c) Poverty index in region of interest under `Admin` partition. (d) Crime count in region of interest under `DQN` partition. (e) Poverty index in region of interest under `DQN` partition.

three communities, found in Figure 4.6(c). It is clear that under `Admin` partition, these communities are nearly identical in poverty index, while their crime counts differ dramatically. Next, we visualize the crime count and poverty index with `DQN` partition in Figure 4.6(d) and Figure 4.6(e), respectively. We can see that these three regions of interest still exhibit similar poverty levels, but their crime counts are much closer to each other.

The observation above reveals the intuition of `DQN` partition: it attempts to make the spatial distributions of X and Y similar. In other words, this is how `DQN` partition reduces the prediction error.

We also note that community #47 dramatically increased in size under `DQN` partition. Community #47 used to be the smallest community according to the Chicago's administrative boundary partition, with a population of less than 3,000

people. The variance penalty in our objective function (Equation 4.1) causes our proposed method to favor regions that are similar in terms of population. It is likely that this community is expanded to yield better correlation between poverty and crime, as well as balance out the population distribution over communities.

4.6 Conclusion

In this chapter, we proposed a new problem called task-specific region partition. The problem is motivated by the fact that existing administrative boundaries are static regardless of the target variable, and we observed cases where it is necessary to have different partitions for different tasks. The task-specific region partition problem is NP-hard, and hence directly searching for a global optimal is difficult. Three variants of MCMC methods are proposed to solve this combinatorial optimization problem. First, a Naive MCMC that generates the next sample by random sampling. Second, a heuristic-based, Guided MCMC method that prefers to select from community areas with larger errors to generate the next sample. Finally, we employ reinforcement learning to automatically learn a sample strategy. Our methods are evaluated on two prediction tasks, i.e. crime prediction and real estate price prediction. The learned predictions consistently outperform the administrative boundaries in both tasks.

Chapter 5
Non-Stationary Model for Crime Rate Inference Using Modern Urban Data

5.1 Inference Model

5.1.1 Linear Regression

The most straightforward prediction model is linear regression. This model assumes that the error term for y_i follows a Gaussian distribution $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$.

Equation (5.1) gives the linear regression formulation of our problem:

$$\vec{\mathbf{y}} = \alpha^T X^N + \beta^f W^f \vec{\mathbf{y}} + \beta^g W^g \vec{\mathbf{y}} + \epsilon. \tag{5.1}$$

$X^N \in R^{d_N \times n}$ is the nodal feature matrix where column i is the nodal feature vector of region i, d_N is the dimension of nodal features, and n is the number of regions. Both demographic features and POI distribution features are included in X^N as nodal features. $W^f \in \mathbb{R}^{n \times n}$ is the flow matrix of taxi flow, and $W^g \in \mathbb{R}^{n \times n}$ is the spatial matrix representing the geographical adjacency. In addition, $\alpha \in \mathbb{R}^{d_N}$ and $\beta^f, \beta^g \in \mathbb{R}$ are the coefficients for corresponding features. Note that $\epsilon \in \mathbb{R}^n$ is the only stochastic variable on the right-hand side; all other terms are fixed observation values. Therefore, we incorporate all the fixed observations into one

term $X \in \mathbb{R}^{(d_N+2)\times n}$, and we get the standard regression problem:

$$\vec{\mathbf{y}} = \vec{\mathbf{w}}^T X + \epsilon,$$

where $\vec{\mathbf{w}} = [\alpha^T, \beta^f, \beta^g]^T$ is the concatenation of all coefficients.

5.1.2 Negative Binomial Regression

In our problem, we aim to infer the crime rate, which is guaranteed to be a non-negative integer. However, linear regression does not ensure this property. *Poisson regression* is another form of regression, more appropriate for non-negative data than linear regression [56, 57]. With shortened notation $\vec{\mathbf{x}}_i$, which represents all features in a region, the Poisson regression model has the exponential function as link function

$$E(y_i) = e^{\vec{\mathbf{w}}^T \vec{\mathbf{x}}_i}. \tag{5.2}$$

In the following, we omit the index i wherever it is clear to refer to the variable of a single region. The link function comes from the assumption that y follows the Poisson distribution with mean λ. Additionally, the mean λ is determined by observed independent variables $\vec{\mathbf{x}}$, i.e. $\lambda = e^{\vec{\mathbf{w}}^T \vec{\mathbf{x}}}$. Adding all together, the joint probability of y is

$$P(y|\vec{\mathbf{w}}) = \frac{e^{-e^{\vec{\mathbf{w}}^T \vec{\mathbf{x}}}} (e^{\vec{\mathbf{w}}^T \vec{\mathbf{x}}})^y}{y!}. \tag{5.3}$$

However, Poisson regression enforces the mean and variance of dependent variable y to be equal. This restriction leads to the "over-dispersion" issue for some real problems, that is the presence of larger variability in data set than the statistical model expected. To address this, we use the Poisson-Gamma mixture model, which is also known as *negative binomial regression*. Negative binomial regression is frequently used in crime research [58].

Given that the crime rate y follows Poisson distribution with mean λ, in order to allow for larger variance, λ itself is a random variable having a Gamma distribution with shape k and scale $\theta = \frac{p}{1-p}$. The probability density function of y becomes

$$P(y|k,p) = \int_0^\infty P_{Poisson}(y|\lambda) \cdot P_{Gamma}(\lambda|k,p) d\lambda$$

$$= \int_0^\infty \frac{\lambda^y}{y!} e^{-\lambda} \cdot \lambda^{k-1} \frac{e^{-\lambda(1-p)/p}}{(\frac{p}{1-p})^k \Gamma(k)} d\lambda$$
$$= \frac{\Gamma(k+y)}{y!\Gamma(k)} p^y (1-p)^k. \tag{5.4}$$

This is exactly the probability density function of negative binomial distribution.

In negative binomial regression, the link function is

$$E(y) = e^{\vec{\mathbf{w}}^T \vec{\mathbf{x}} + \epsilon}. \tag{5.5}$$

The error term e^ϵ is the mixture prior from the Gamma distribution, and we assume its mean is 1, i.e. $E(e^\epsilon) = 1$. This setting ensures that $E(y) = e^{\vec{\mathbf{w}}^T \vec{\mathbf{x}}} \cdot e^\epsilon = e^{\vec{\mathbf{w}}^T \vec{\mathbf{x}}}$. Meanwhile, given the probability density function of negative binomial distribution in Equation (5.4), the mean of negative binomial distribution is $\frac{pk}{1-p}$. Combining the theoretical mean with the link function, we have $p = \frac{e^{\vec{\mathbf{w}}^T \vec{\mathbf{x}}}}{e^{\vec{\mathbf{w}}^T \vec{\mathbf{x}}} + k}$. Therefore, the probability mass function of y becomes

$$P(y|\vec{\mathbf{w}}, k) = \frac{\Gamma(k+y)}{y!\Gamma(k)} \left(\frac{e^{\vec{\mathbf{w}}^T \vec{\mathbf{x}}}}{e^{\vec{\mathbf{w}}^T \vec{\mathbf{x}}} + k} \right)^y \left(\frac{k}{e^{\vec{\mathbf{w}}^T \vec{\mathbf{x}}} + k} \right)^k. \tag{5.6}$$

The log-likelihood function of negative binomial model is given in Equation (5.7), where w and θ can be estimated by maximizing likelihood.

$$\mathcal{L}(\vec{\mathbf{w}}, k; \vec{\mathbf{y}}, X) = \sum_{i=1}^{n} \left\{ y_i \ln \left(\frac{e^{\vec{\mathbf{w}}^T \vec{\mathbf{x}}_i}}{e^{\vec{\mathbf{w}}^T \vec{\mathbf{x}}_i} + k} \right) + k \ln \left(\frac{k}{e^{\vec{\mathbf{w}}^T \vec{\mathbf{x}}_i} + k} \right) \right.$$
$$\left. + \ln \Gamma(y_i + k) - \ln \Gamma(y_i + 1) - \ln \Gamma(k) \right\}. \tag{5.7}$$

5.1.3 Non-Stationary Model

The two regression models described above assume the statistical correlations between crime rate and observed features are constant over space, because they learn one set of parameters for all community areas. In the real world, it is very likely that some statistical correlations between crime rate and observed features are not stationary over space. In this section we propose to apply a non-stationary model, called geographically weighted regression (GWR) [**?**], to capture the different

crime correlations at different places.

Formally, a global spatial regression model such as the aforementioned two models has the following form

$$y = f(\vec{\mathbf{x}}, \vec{\mathbf{w}}), \tag{5.8}$$

where $\vec{\mathbf{w}}$ is the parameter of the regression function f. Given a set of data points $\{y_i, \vec{\mathbf{x}}_i\}_{i=1}^n$ sampled at locations $l_1, \cdots, l_n$, the maximum likelihood estimation of parameter $\vec{\mathbf{w}}$ is given by

$$\vec{\mathbf{w}}^* = \arg\max_{\vec{\mathbf{w}}} \sum_{i=1}^{n} \mathcal{L}(y_i, f(\vec{\mathbf{x}}_i, \vec{\mathbf{w}})). \tag{5.9}$$

This global model is stationary, because the weights used for predictions are the same at all locations, when we fit the model to find the optimal parameter.

Instead, the GWR learns a *local* regression function f with parameter $\vec{\mathbf{w}}_i$ at each location of interest l_i:

$$y = f(\vec{\mathbf{x}}, \vec{\mathbf{w}}_i), \quad \forall l_i \in \{l_1, l_2, \cdots, l_n\}, \tag{5.10}$$

where l_i is usually a geospatial coordinate in the two dimensional space. In order to train a lot of local models, we need a larger number of samples at each location l_i, which are usually not available. To address this issue, GWR uses the spatially nearby samples and gives each sample a weight according to the distance between sample point and target location l_i. The objective for the local model at location l_i is

$$\vec{\mathbf{w}}_i^* = \arg\min_{\vec{\mathbf{w}}_i} \sum_{j=1}^{n} \gamma_{ij} \mathcal{L}\left(y_j, f(\vec{\mathbf{x}}_j, \vec{\mathbf{w}}_i)\right), \tag{5.11}$$

where γ_{ij} is the spatial kernel to weight the neighboring data point at location l_j for regression model at location l_i.

Choice of spatial kernel γ. There are several spatial kernels we can choose from. The most straightforward solution is to exclude samples that are further away from target location. Namely,

$$\gamma_{ij} = \begin{cases} 1 & \text{if } d_{ij} < \tau \\ 0 & \text{otherwise,} \end{cases} \tag{5.12}$$

where d_{ij} is the distance between l_i and l_j, and τ is a distance threshold. Clearly, such a solution suffers from the discontinuity.

A better solution is to specify the weight γ as a continuous function of distance d, which is

$$\gamma_{ij} = \exp\left(-\frac{d_{ij}^2}{2h^2}\right), \tag{5.13}$$

where h is referred to as the bandwidth of the Gaussian kernel. Intuitively, when the samples are dense near the target location l_t, the h can be set smaller, so that we give lower weights to those samples far away. On the other hand, if the samples are sparse, h should be set larger, so that we consider those further away samples as well to train our model. When h is set to infinity, the GWR becomes a global model, since all samples have equal weight 1.

One issue with the Gaussian kernel in Equation (5.13) is that when the samples are dense, it over-smooths local models by considering too many samples at each location. A popular alternative kernel utilizes the bi-square function,

$$\gamma_{ij} = \begin{cases} \left(1 - \frac{d_{ij}^2}{\tau^2}\right)^2 & \text{if } d_{ij} < \tau \\ 0 & \text{otherwise.} \end{cases} \tag{5.14}$$

The bi-square kernel functions provides continuous weight for samples up to distance τ. In our problem, since we do not have too many samples available, therefore we use the Gaussian kernel, and in experiment we will show the bandwidth tuning to get the best results.

Applying GWR on existing methods. The GWR is more like a framework rather than a method, which can be applied to many existing regression methods. The classic GWR is applied to linear regression model resulting in the following objective for location l_i

$$\vec{\mathbf{w}}_i^* = \arg\min_{w_i} \sum_{j=1}^{n} \gamma_{ij}(y_j - \vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_j)^2. \tag{5.15}$$

Similarly, the GWR framework can be employed with the negative binomial regression model, and we call this *geographically weighted negative binomial regression* (GWNBR). Here, the objective for model at location l_i is to optimize the

weighted log-likelihood function:

$$\mathcal{L}(\vec{\mathbf{w}}_i, k_i; \vec{\mathbf{y}}, X) = \sum_{j=1}^{n} \gamma_{ij} \Bigg\{ y_j \ln \left(\frac{e^{\vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_j}}{e^{\vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_j} + k_i} \right) + k_i \ln \left(\frac{k_i}{e^{\vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_j} + k_i} \right) + \ln \Gamma(y_j + k_i) - \ln \Gamma(y_j + 1) - \ln \Gamma(k_i) \Bigg\}. \quad (5.16)$$

5.1.4 Optimization

The objective in Equation (5.16) can be solved using a block coordinate gradient descent method, by alternatively solving $\vec{\mathbf{w}}_i$ and k_i. Details for solving each step are given below.

Fix k_i, solve $\vec{\mathbf{w}}_i$:

When k_i is fixed, the objective function can be simplified as follows:

$$\min_{\vec{\mathbf{w}}_i} - \sum_{j=1}^{n} \gamma_{ij} \Bigg\{ y_j \ln \left(\frac{e^{\vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_j}}{e^{\vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_j} + k_i} \right) + k_i \ln \left(\frac{k_i}{e^{\vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_j} + k_i} \right) \Bigg\}. \quad (5.17)$$

The gradient is:

$$\frac{\partial \mathcal{L}}{\partial \vec{\mathbf{w}}_i} = \sum_{i=1}^{n} \gamma_{ij} \{ y_i \frac{\vec{\mathbf{x}}_j e^{\vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_j}}{e^{\vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_j}} - \frac{(y_i + k_i) \vec{\mathbf{x}}_j e^{\vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_j}}{e^{\vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_j} + k_i} \}. \quad (5.18)$$

Then,

$$\vec{\mathbf{w}}_i^t = \vec{\mathbf{w}}_i^{t-1} + \alpha \frac{\partial \mathcal{L}}{\partial \vec{\mathbf{w}}_i^{t-1}}. \quad (5.19)$$

Fix $\vec{\mathbf{w}}_i$, solve k_i:

When $\vec{\mathbf{w}}_i$ is fixed, the objective function becomes:

$$\min_{k_i} = - \sum_{j=1}^{n} \gamma_{ij} \Bigg\{ y_j \ln \left(\frac{e^{\vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_j}}{e^{\vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_j} + k_i} \right) + k_i \ln \left(\frac{k_i}{e^{\vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_j} + k_i} \right) + \ln \Gamma(y_j + k_i) - \ln \Gamma(k_i) \Bigg\}. \quad (5.20)$$

The gradient is:

$$\frac{\partial \mathcal{L}}{\partial k_i} = \sum_{i=1}^{n} \gamma_{ij} \{ \frac{y_i}{e^{\vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_j} + k_i} + \ln(\frac{k_i}{e^{\vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_j} + k_i}) - \frac{k_i}{e^{\vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_j} + k_i} + 1 + \psi(y_j + k_i) - \psi(k_i) + y_i \}, \quad (5.21)$$

where $\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$ is digamma function. Then,

$$k_i^t = k_i^{t-1} + \alpha \frac{\partial \mathcal{L}}{\partial k_i^{t-1}}. \quad (5.22)$$

Chapter 6
Discussion

In this chapter, we discuss some potential topics as future work of this dissertation. Following the three challenges in this dissertation, there are potential improvements on each of the topic.

6.1 Supervised Embedding Method

The region embedding in Chapter 3 improves the performance of prediction tasks. However, the improvement is still limited. We found the main reason is that the embedding of region is learned in an unsupervised fashion. A potential improvement is to keep the task in mind, and devise some unsupervised embedding learning technique, which potentially will significantly improve the performance.

6.2 General Region Partition

One of the drawbacks of task-specific region partition is that the region partition is not stable over multiple runs of the same algorithm. The reason is that the proposed methods are stochastic methods. To further improve the stability of the identified partition, we are thinking using multiple tasks and jointly learning the partition. Another benefit of this joint partition learning is that the learned partition is representative and suitable for multiple tasks.

6.3 Adaptive Local Model

The GWR method is a simple strategy to design local models. However, we notice that the temporal features are not incorporated within the GWR model. Ideally, we want to build local model to account for both temporal change and spatial correlations. Further more, we do not want to build a lot of independent local models, because there are still correlations across the space. A viable solution is to jointly build a lot of local models. Namely, the local models share certain global structure, meanwhile each local model is adaptive to its local property.

Bibliography

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

[2] Whong, C., "Foiling NYC's Taxi Trip Data: `http://chriswhong.com/open-data/foil_nyc_taxi/`," .

[3] Haynes, K. E. and A. S. Fotheringham (1984) *Gravity and spatial interaction models*, vol. 2, Sage publications Beverly Hills.

[4] Rodrigue, J.-P., C. Comtois, and B. Slack (2013) *The geography of transport systems*, Routledge.

[5] Mátyás, L. (1997) "Proper econometric specification of the gravity model," *The world economy*, **20**(3), pp. 363–368.

[6] Anselin, L. (1980) *Estimation methods for spatial autoregressive structures: A study in spatial econometrics*, vol. 8, Program in Urban and Regional Studies, Cornell University.

[7] Carrere, C. (2006) "Revisiting the effects of regional trade agreements on trade flows with proper specification of the gravity model," *European Economic Review*, **50**(2), pp. 223–247.

[8] Egger, P. and M. Pfaffermayr (2003) "The proper panel econometric specification of the gravity equation: A three-way model with bilateral interaction effects," *Empirical Economics*, **28**(3), pp. 571–580.

[9] Martínez-Zarzoso, I., F. Nowak-Lehmann, et al. (2003) "Augmented gravity model: An empirical application to Mercosur-European Union trade flows," *Journal of applied economics*, **6**(2), pp. 291–316.

[10] Hanski, I., M. Kuussaari, and M. Nieminen (1994) "Metapopulation structure and migration in the butterfly Melitaea cinxia," *Ecology*, **75**(3), pp. 747–762.

[11] Karemera, D., V. I. Oguledo, and B. Davis (2000) "A gravity model analysis of international migration to North America," *Applied Economics*, **32**(13), pp. 1745–1755.

[12] Lewer, J. J. and H. Van den Berg (2008) "A gravity model of immigration," *Economics letters*, **99**(1), pp. 164–167.

[13] Jung, W.-S., F. Wang, and H. E. Stanley (2008) "Gravity model in the Korean highway," *EPL (Europhysics Letters)*, **81**(4), p. 48005.

[14] Roughan, M., A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang (2002) "Experience in measuring backbone traffic variability: Models, metrics, measurements and meaning," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, ACM, pp. 91–92.

[15] Khadaroo, J. and B. Seetanah (2008) "The role of transport infrastructure in international tourism development: A gravity model approach," *Tourism management*, **29**(5), pp. 831–840.

[16] Krings, G., F. Calabrese, C. Ratti, and V. D. Blondel (2009) "Urban gravity: a model for inter-city telecommunication flows," *Journal of Statistical Mechanics: Theory and Experiment*, **2009**(07), p. L07003.

[17] Fischer, M. M. and S. Gopal (1994) "ARTIFICIAL NEURAL NETWORKS: A NEW APPROACH TO MODELING INTERREGIONAL TELECOMMUNICATION FLOWS*," *Journal of Regional Science*, **34**(4), pp. 503–527.

[18] Black, W. R. (1995) "Spatial interaction modeling using artificial neural networks," *Journal of Transport Geography*, **3**(3), pp. 159–166.

[19] Anselin, L., J. Cohen, D. Cook, W. Gorr, and G. Tita (2000) "Spatial analyses of crime," *Criminal justice*, **4**(2), pp. 213–262.

[20] Kakamu, K., W. Polasek, and H. Wago (2008) "Spatial interaction of crime incidents in Japan," *Mathematics and Computers in Simulation*, **78**(2), pp. 276–282.

[21] Browning, C. R., R. D. Dietz, and S. L. Feinberg (2004) "The paradox of social organization: Networks, collective efficacy, and violent crime in urban neighborhoods," *Social Forces*, **83**(2), pp. 503–534.

[22] LeSage, J. P., M. M. Fischer, and T. Scherngell (2007) "Knowledge spillovers across Europe: Evidence from a Poisson spatial interaction model with spatial effects*," *Papers in Regional Science*, **86**(3), pp. 393–421.

[23] Fischer, M. M., T. Scherngell, and E. Jansenberger (2006) "The Geography of Knowledge Spillovers Between High-Technology Firms in Europe: Evidence from a Spatial Interaction Modeling Perspective," *Geographical Analysis*, **38**(3), pp. 288–309.

[24] Huang, Z., A. J. Tatem, et al. (2013) "Global malaria connectivity through air travel," *Malar J*, **12**(1), p. 269.

[25] Tatem, A. J. (2014) "Mapping population and pathogen movements," *International health*, p. ihu006.

[26] Pindolia, D. K., A. J. Garcia, Z. Huang, D. L. Smith, V. A. Alegana, A. M. Noor, R. W. Snow, and A. J. Tatem (2013) "The demographics of human and malaria movement and migration patterns in East Africa," *Malar J*, **12**(397), pp. 10–1186.

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

[28] Yuan, J., Y. Zheng, and X. Xie (2012) "Discovering regions of different functions in a city using human mobility and POIs," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 186–194.

[29] Berlingerio, M., F. Calabrese, G. Di Lorenzo, R. Nair, F. Pinelli, and M. L. Sbodio (2013) "AllAboard: a system for exploring urban mobility and optimizing public transport using cellphone data," in *Machine learning and knowledge discovery in databases*, Springer, pp. 663–666.

[30] Pontes, T., M. Vasconcelos, J. Almeida, P. Kumaraguru, and V. Almeida (2012) "We Know Where You Live: Privacy Characterization of Foursquare Behavior," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, ACM, New York, NY, USA, pp. 898–905.
URL `http://doi.acm.org/10.1145/2370216.2370419`

[31] Li, R., S. Wang, H. Deng, R. Wang, and K. C.-C. Chang (2012) "Towards Social User Profiling: Unified and Discriminative Influence Model for Inferring Home Locations," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, ACM, New York, NY, USA, pp. 1023–1031.
URL `http://doi.acm.org/10.1145/2339530.2339692`

[32] Dey, R., C. Tang, K. Ross, and N. Saxena (2012) "Estimating age privacy leakage in online social networks," in *INFOCOM, 2012 Proceedings IEEE*, pp. 2836–2840.

[33] Mislove, A., B. Viswanath, K. P. Gummadi, and P. Druschel (2010) "You Are Who You Know: Inferring User Profiles in Online Social Networks," in *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, ACM, New York, NY, USA, pp. 251–260. URL `http://doi.acm.org/10.1145/1718487.1718519`

[34] Fortunato, S. (2010) "Community detection in graphs," *Physics reports*, **486**(3), pp. 75–174.

[35] Guille, A., H. Hacid, C. Favre, and D. A. Zighed (2013) "Information diffusion in online social networks: A survey," *ACM SIGMOD Record*, **42**(2), pp. 17–28.

[36] Breese, J. S., D. Heckerman, and C. Kadie (1998) "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc., pp. 43–52.

[37] Rodriguez, M. G., D. Balduzzi, and B. Schölkopf (2011) "Uncovering the temporal dynamics of diffusion networks," *arXiv preprint arXiv:1105.0697*.

[38] Gomez Rodriguez, M., J. Leskovec, and A. Krause (2010) "Inferring networks of diffusion and influence," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 1019–1028.

[39] Pan, W., N. Aharony, and A. Pentland (2011) "Composite social network for predicting mobile apps installation," *arXiv preprint arXiv:1106.0359*.

[40] Madan, A., K. Farrahi, D. Gatica-Perez, and A. S. Pentland (2011) "Pervasive sensing to model political opinions in face-to-face networks," in *Pervasive Computing*, Springer, pp. 214–231.

[41] Zhong, E., W. Fan, J. Wang, L. Xiao, and Y. Li (2012) "Comsoc: adaptive transfer of user behaviors over composite social network," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 696–704.

[42] Baum, K. (2005) *Juvenile victimization and offending, 1993-2003*, US Department of Justice, Office of Justice Programs, Bureau of Justice Statistics.

[43] Finkelhor, D. (2008) *Childhood victimization: violence, crime, and abuse in the lives of young people: violence, crime, and abuse in the lives of young people*, Oxford University Press, USA.

[44] for Disease Control, N. C. and P. (CDC) (2015) "Leading Causes of Nonfatal Injury, United States 2001 - 2013." *Injury Prevention and Control: data and statistics*.

[45] Graif, C. (2015) "Toward a Geographically Extended Perspective of Neighborhood Effects on Children's Victimization," *American Society of Criminology Annual Meeting*.

[46] Tribune, C. (2015), "A tale of 3 cities: LA and NYC outpace Chicago in curbing violence," .
URL http://www.chicagotribune.com/news/ct-violence-chicago-new-york-los-angeles-met-20150918-story.html

[47] Graif, C. and R. J. Sampson (2009) "Spatial heterogeneity in the effects of immigration and diversity on neighborhood homicide rates," *Homicide Studies*.

[48] Anselin, L. (2002) "Under the hood: issues in the specification and interpretation of spatial regression models," *Agricultural economics*, **27**(3), pp. 247–267.

[49] Zheng, Y., L. Capra, O. Wolfson, and H. Yang (2014) "Urban computing: concepts, methodologies, and applications," *ACM TIST*, **5**(3), p. 38.

[50] Yuan, J., Y. Zheng, and X. Xie (2012) "Discovering regions of different functions in a city using human mobility and POIs," in *ACM SIGKDD*, ACM, pp. 186–194.

[51] Graif, C., A. S. Gladfelter, and S. A. Matthews (2014) "Urban poverty and neighborhood effects on crime: Incorporating spatial and network perspectives," *Sociology Compass*, **8**(9), pp. 1140–1155.

[52] Cohen, L. E. and M. Felson (1979) "Social change and crime rate trends: A routine activity approach," *American sociological review*, pp. 588–608.

[53] Brantingham, P. and P. Brantingham (1995) "Criminality of place," *European journal on criminal policy and research*, **3**(3), pp. 5–26.

[54] Wikipedia (2015), "Community areas in Chicago — Wikipedia, The Free Encyclopedia," .
URL https://en.wikipedia.org/w/index.php?title=Community_areas_in_Chicago&oldid=696795849

[55] (2015), "City of Chicago data portal," https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2.

[56] Gardner, W., E. P. Mulvey, and E. C. Shaw (1995) "Regression analyses of counts and rates: Poisson, overdispersed Poisson, and negative binomial models." *Psychological bulletin*, **118**(3), p. 392.

[57] Lambert, D. (1992) "Zero-inflated Poisson regression, with an application to defects in manufacturing," *Technometrics*, **34**(1), pp. 1–14.

[58] Osgood, D. W. (2000) "Poisson-based regression analysis of aggregate crime rates," *Journal of quantitative criminology*, **16**(1), pp. 21–43.

[59] Bogomolov, A., B. Lepri, J. Staiano, N. Oliver, F. Pianesi, and A. Pentland (2014) "Once upon a crime: towards crime prediction from demographics and mobile data," in *Proceedings of the 16th international conference on multimodal interaction*, ACM, pp. 427–434.

[60] Hsieh, C.-C. and M. D. Pugh (1993) "Poverty, income inequality, and violent crime: a meta-analysis of recent aggregate data studies," *Criminal Justice Review*, **18**(2), pp. 182–202.

[61] Wolfe, M. K. and J. Mennis (2012) "Does vegetation encourage or suppress urban crime? Evidence from Philadelphia, PA," *Landscape and Urban Planning*, **108**(2), pp. 112–122.

[62] Sahbaz, O. and B. Hillier (2007) "The story of the crime: functional, temporal and spatial tendencies in street robbery," in *Proc of 6th International Space Syntax Symposium, Istanbul*, pp. 4–14.

[63] Jacobs, J. (1961) *The death and life of great American cities*, Vintage.

[64] "United States Census Bureau," http://www.census.gov.

[65] Sampson, R. J., S. W. Raudenbush, and F. Earls (1997) "Neighborhoods and violent crime: A multilevel study of collective efficacy," *Science*, **277**(5328), pp. 918–924.

[66] "Foursquare Venues Service," https://developer.foursquare.com/overview/venues.html.

[67] Gorman, D. M., P. W. Speer, P. J. Gruenewald, and E. W. Labouvie (2001) "Spatial dynamics of alcohol availability, neighborhood structure and violent crime." *Journal of studies on alcohol*, **62**(5), pp. 628–636.

[68] Burnell, J. D. (1988) "Crime and racial composition in contiguous communities as negative externalities: prejudiced household's evaluation of crime rate and segregation nearby reduces housing values and tax revenues," *American Journal of Economics and Sociology*, **47**(2), pp. 177–193.

[69] Anselin, L., J. Cohen, D. Cook, W. Gorr, and G. Tita (2000) "Spatial analyses of crime," *Criminal justice*, **4**(2), pp. 213–262.

[70] Morenoff, J. D. and R. J. Sampson (1997) "Violent crime and the spatial dynamics of neighborhood transition: Chicago, 1970–1990," *Social forces*, **76**(1), pp. 31–64.

[71] Mohler, G. O., M. B. Short, P. J. Brantingham, F. P. Schoenberg, and G. E. Tita (2012) "Self-exciting point process modeling of crime," *Journal of the American Statistical Association*.

[72] Wang, T., C. Rudin, D. Wagner, and R. Sevieri (2013) "Learning to detect patterns of crime," in *Machine Learning and Knowledge Discovery in Databases*, Springer.

[73] Ehrlich, I. (1975) "On the relation between education and crime," in *Education, income, and human behavior*, NBER, pp. 313–338.

[74] Braithwaite, J. (1989) *Crime, shame and reintegration*, Cambridge University Press.

[75] Patterson, E. B. (1991) "Poverty, income inequality, and community crime rates," *Criminology*, **29**(4), pp. 755–776.

[76] Freeman, R. B. (1999) "The economics of crime," *Handbook of labor economics*, **3**, pp. 3529–3571.

[77] Wang, X., M. S. Gerber, and D. E. Brown (2012) "Automatic crime prediction using events extracted from twitter posts," in *Social Computing, Behavioral-Cultural Modeling and Prediction*, Springer, pp. 231–238.

[78] Gerber, M. S. (2014) "Predicting crime using Twitter and kernel density estimation," *Decision Support Systems*, **61**.

[79] Traunmueller, M., G. Quattrone, and L. Capra (2014) "Mining mobile phone data to investigate urban crime theories at scale," in *Social Informatics*, Springer, pp. 396–411.

[80] Ratcliffe, J. H. (2006) "A temporal constraint theory to explain opportunity-based spatial offending patterns," *Journal of Research in Crime and Delinquency*, **43**(3), pp. 261–291.

[81] Toole, J. L., N. Eagle, and J. B. Plotkin (2011) "Spatiotemporal correlations in criminal offense records," *ACM Transactions on Intelligent Systems and Technology (TIST)*, **2**(4), p. 38.

[82] Short, M. B., M. R. D'ORSOGNA, V. B. Pasour, G. E. Tita, P. J. Brantingham, A. L. Bertozzi, and L. B. Chayes (2008) "A statistical model of criminal behavior," *Mathematical Models and Methods in Applied Sciences*, **18**(supp01), pp. 1249–1267.

[83] Chainey, S., L. Tompson, and S. Uhlig (2008) "The utility of hotspot mapping for predicting spatial patterns of crime," *Security Journal*, **21**(1), pp. 4–28.

[84] Eck, J., S. Chainey, J. Cameron, and R. Wilson (2005) "Mapping crime: Understanding hotspots," .

[85] Nakaya, T. and K. Yano (2010) "Visualising Crime Clusters in a Space-time Cube: An Exploratory Data-analysis Approach Using Space-time Kernel Density Estimation and Scan Statistics," *Transactions in GIS*, **14**(3), pp. 223–239.

[86] Buczak, A. L. and C. M. Gifford (2010) "Fuzzy association rule mining for community crime pattern discovery," in *ACM SIGKDD Workshop on Intelligence and Security Informatics*, ACM, p. 2.

[87] Nations, U. (2014) "World Urbanization Prospects: The 2014 Revision, Highlights. Department of Economic and Social Affairs," *Population Division, United Nations*.

[88] Digital, C. (2016), "Chicago Taxi Data Released," `http://digital.cityofchicago.org/index.php/chicago-taxi-data-released/`, accessed: November, 2016.

[89] Wang, H., D. Kifer, C. Graif, and Z. Li (2016) "Crime Rate Inference with Big Data," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, ACM, New York, NY, USA, pp. 635–644.
URL `http://doi.acm.org/10.1145/2939672.2939736`

[90] Perozzi, B., R. Al-Rfou, and S. Skiena (2014) "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 701–710.

[91] Tang, J., M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei (2015) "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*, ACM, pp. 1067–1077.

[92] Qi, G., X. Li, S. Li, G. Pan, Z. Wang, and D. Zhang (2011) "Measuring social functions of city regions from large-scale taxi behaviors," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, IEEE, pp. 384–388.

[93] Pan, G., G. Qi, Z. Wu, D. Zhang, and S. Li (2013) "Land-use classification using taxi GPS traces," *IEEE Transactions on Intelligent Transportation Systems*, **14**(1), pp. 113–123.

[94] Zheng, Y., F. Liu, and H.-P. Hsieh (2013) "U-air: When urban air quality inference meets big data," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 1436–1444.

[95] Zheng, Y., T. Liu, Y. Wang, Y. Zhu, Y. Liu, and E. Chang (2014) "Diagnosing New York city's noises with ubiquitous data," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ACM, pp. 715–725.

[96] Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013) "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119.

[97] Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013) "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*.

[98] Mikolov, T., W.-t. Yih, and G. Zweig (2013) "Linguistic Regularities in Continuous Space Word Representations." in *HLT-NAACL*, vol. 13, pp. 746–751.

[99] Pennington, J., R. Socher, and C. D. Manning (2014) "Glove: Global Vectors for Word Representation." in *EMNLP*, vol. 14, pp. 1532–43.

[100] Grover, A. and J. Leskovec (2016) "node2vec: Scalable Feature Learning for Networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM.

[101] Nekola, J. C. and P. S. White (1999) "The distance decay of similarity in biogeography and ecology," *Journal of Biogeography*, **26**(4), pp. 867–878.

[102] Walker, A. J. (1974) "New fast method for generating discrete random numbers with arbitrary frequency distributions," *Electronics Letters*, **10**(8), pp. 127–128.

[103] Bureau, U. S. C. (2010), "Demographics survey," `http://www.census.gov`.

[104] Service, F. V. (2015), `https://developer.foursquare.com/overview/venues.html`.

[105] of Chicago data portal, C. (2015), `https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2`.

[106] Zillow.com (2017), "Real Estate Value in Chicago," `https://www.zillow.com/`.

[107] Fu, Y., Y. Ge, Y. Zheng, Z. Yao, Y. Liu, H. Xiong, and J. Yuan (2014) "Sparse real estate ranking with online user reviews and offline moving behaviors," in *Data Mining (ICDM), 2014 IEEE International Conference on*, IEEE, pp. 120–129.

[108] Cox, T. F. and M. A. Cox (2000) *Multidimensional scaling*, CRC press.

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

[110] Belkin, M. and P. Niyogi (2001) "Laplacian eigenmaps and spectral techniques for embedding and clustering." in *NIPS*, vol. 14, pp. 585–591.

[111] Ahmed, A., N. Shervashidze, S. Narayanamurthy, V. Josifovski, and A. J. Smola (2013) "Distributed large-scale natural graph factorization," in *Proceedings of the 22nd international conference on World Wide Web*, ACM, pp. 37–48.

[112] Wang, S., J. Tang, C. Aggarwal, and H. Liu (2016) "Linked Document Embedding for Classification," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, ACM, pp. 115–124.

[113] Xie, M., H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang (2016) "Learning Graph-based POI Embedding for Location-based Recommendation," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, ACM, pp. 15–24.

[114] Team, O. D. (2018), "NYC Open Data Dashboard," .
URL `https://opendata.cityofnewyork.us/dashboard/`

[115] Wang, H., H. Yao, D. Kifer, C. Graif, and Z. Li (2017) "Non-Stationary Model for Crime Rate Inference Using Modern Urban Data," *IEEE Transactions on Big Data*.

[116] Miller, H. J. and J. Han (2009) *Geographic data mining and knowledge discovery*, CRC Press.

[117] Wang, H. and Z. Li (2017) "Region Representation Learningvia Mobility Flow," in *In Proceedings of CIKM'17*, CIKM'17, p. 10 pages.

[118] Wang, H., Y.-H. Kuo, D. Kifer, and Z. Li (2016) "A simple baseline for travel time estimation using large-scale trip data," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, p. 61.

[119] Wu, F., H. Wang, and Z. Li (2016) "Interpreting traffic dynamics using ubiquitous urban data," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, p. 69.

[120] Li, Y., Y. Zheng, H. Zhang, and L. Chen (2015) "Traffic prediction in a bike-sharing system," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, p. 33.

[121] Xu, F., Y. Li, H. Wang, P. Zhang, and D. Jin (2017) "Understanding mobile traffic patterns of large scale cellular towers in urban environment," *IEEE/ACM Transactions on Networking (TON)*, **25**(2), pp. 1147–1161.

[122] Zheng, Y., X. Yi, M. Li, R. Li, Z. Shan, E. Chang, and T. Li (2015) "Forecasting fine-grained air quality based on big data," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 2267–2276.

[123] Strens, M. (2003) "Evolutionary MCMC sampling and optimization in discrete spaces," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 736–743.

[124] Li, K. and J. Malik (2016) "Learning to optimize," *arXiv preprint arXiv:1606.01885*.

[125] Andreev, K. and H. Racke (2006) "Balanced graph partitioning," *Theory of Computing Systems*, **39**(6), pp. 929–939.

[126] Andrieu, C., N. De Freitas, A. Doucet, and M. I. Jordan (2003) "An introduction to MCMC for machine learning," *Machine learning*, **50**(1-2), pp. 5–43.

[127] Murphy, K. P. (2012) *Machine Learning: A Probabilistic Perspective*, The MIT Press, Cambridge, Massachusetts.

[128] Sutton, R. S. and A. G. Barto (1998) *Reinforcement learning: An introduction*, vol. 1, MIT press Cambridge.

[129] Van Hasselt, H., A. Guez, and D. Silver (2016) "Deep Reinforcement Learning with Double Q-Learning." in *AAAI*, vol. 16, pp. 2094–2100.

[130] (2010), "United States Census Bureau. Demographics Survey," .
URL `http://www.census.gov`

Vita

Hongjian Wang

I get my Ph.D. in Information Sciences and Technology from Pennsylvania State University. My advisor is Prof. Zhenhui (Jessie) Li. My research Interest is data mining, especially mining the spatiotemporal data.

Selected Publications

- **Hongjian Wang**, Porter Jenkins, Hua Wei, Fei Wu, Zhenhui Li. "Learning Task-Specific City Region Partition". In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining* (**KDD 2018**), under review.
- **Hongjian Wang**, Qi Li, Lanbo Zhang, Yue Lu, Zhenhui Li. "Counting Feature Key Selection for CTR Prediction". In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining* (**KDD 2018**), under review.
- **Hongjian Wang**, Huaxiu Yao, Daniel Kifer, Corina Graif, Zhenhui Li. "Non-Stationary Model for Crime Rate Inference Using Modern Urban Data". In *IEEE Transactions on Big Data* (**TBD 2017**).
- **Hongjian Wang**, Zhenhui Li. "Region Representation Learning via Mobility Flow". In *Proceedings of International Conference on Information and Knowledge Management* (**CIKM 2017**).
- **Hongjian Wang**, Zhenhui Li, Yu-Hsuan Kuo, Daniel Kifer. "A Simple Baseline for Travel Time Estimation using Large-Scale Trip Data". In *Proceedings of the 24th ACM International Conference on Advances in Geographic Information Systems* (**SIGSPATIAL 2016**). Short paper.
- **Hongjian Wang**, Daniel Kifer, Corina Graif, Zhenhui Li. "Crime Rate Inference with Big Data". In *Proceedings of the 22nd ACM Conference on Knowledge Discovery and Data Mining* (**KDD 2016**).
- **Hongjian Wang**, Zhenhui Li, Wang-Chien Lee. "PGT: Measuring Mobility Relationship Using Personal, Global and Temporal Factors". In *Proceedings of the 2014 IEEE International Conference on Data Mining* (**ICDM 2014**).
- **Hongjian Wang**, Yamin Zhu, Qian Zhang. "Compressive Sensing based Monitoring with Vehicular Networks". In *Proceedings of the 32nd IEEE International Conference on Computer Communications* (**INFOCOM 2013**).