Title
Social-enabled Urban Data Analytics

Author
Zhang, Danqing

Publication Date
2018

Social-enabled Urban Data Analytics

by

Danqing Zhang

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Civil & Environmental Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Assistant Professor Alexei Pozdnukhov, Chair
Professor Joan Walker
Professor Gregory Biging

Spring 2018

Social-enabled Urban Data Analytics

Abstract

Social-enabled Urban Data Analytics

by

Danqing Zhang

Doctor of Philosophy in Engineering - Civil & Environmental Engineering

University of California, Berkeley

Assistant Professor Alexei Pozdnukhov, Chair

Increasing traffic congestion, vehicle emissions and commuters delay have been major challenges for urban transportation systems for years. The economic cost of traffic congestion in the US is Increasing from 200 billion in 2013 to 293 billion in 2030. There is an increasing need for a better solution to long-term transportation demand forecasting for urban infrastructure planning, and solution to short-term traffic prediction for managing existing urban infrastructure. Accordingly, understanding how urban systems operate and evolve through modeling individuals' daily urban activities has been a major focus of transportation planners, urban planners, and geographers. Traffic data (loop sensors, surveillance cameras, and GPS in taxis, buses), survey data (ACS, CHTS), mobile phone signals (CDR and GPS) and Location Based Social Network (LBSN) data (Facebook, Twitter, Yelp, and Foursquare) have enabled data-driven research on transportation behavior research. The data-driven research, urban data analytics, is an interdisciplinary field where machine learning/ deep learning methods from computer science and optimization/ simulation methods from operation research are applied in conventional city-related fields using spatial-temporal data. In this dissertation, we aim to add the third dimension, social, to urban data analytics research using social-spatial-temporal data, whose key topic is understanding how friendship influences human behavior over time and space. In this era of transformative mobility, this can help better design policies and investment strategies for managing existing urban infrastructure and forecasting future urban infrastructure planning. In this dissertation, we explored two research directions on social-enabled urban data analytics. First, we developed new machine learning models for social discrete choice model, bridging the gap between discrete choice modeling research and computer science research. Second, we developed a methodology framework for synthetic population synthesis using both small data and big data.

The first part of the dissertation focus on modeling social influence on human behavior from a graph modeling perspective, while conforming to the discrete choice modeling framework. The proposed models can be used to model how friends influence individual's travel mode choice and other transportation related choices, which is important to transportation demand forecasting. We propose two novel models with scalable training algorithms: local logistics

graph regularization (LLGR) and latent class graph regularization (LCGR) models. We add social regularization to represent similarity between friends, and we introduce latent classes to account for possible preference discrepancies between different social groups. Training of the LLGR model is performed using alternating direction method of multipliers (ADMM), and training of the LCGR model is performed using a specialized Monte Carlo expectation maximization (MCEM) algorithm. Scalability to large graphs is achieved by parallelizing computation in both the expectation and the maximization steps. The LCGR model is the first latent class classification model that incorporates social relationships among individuals represented by a given graph. To evaluate our two models, we consider three classes of data: small synthetic data to illustrate the knobs of the method, small real data to illustrate one social science use case, and large real data to illustrate a typical large-scale use case in the internet and social media applications. We experiment on synthetic datasets to empirically explain when the proposed model is better than vanilla classification models that do not exploit graph structure. We illustrate how the graph structure and labels, assigned to each node of the graph, need to satisfy certain reasonable properties. We also experiment on real-world data, including both small scale and large scale real-world datasets, to demonstrate on which types of datasets our model can be expected to outperform state-of-the-art models.

This dissertation also develops an algorithmic procedure to incorporate social information into population synthesizer, which is an essential step to incorporate social information into the transportation simulation framework. Agent-based modeling in transportation problems requires detailed information on each of the agents that represent the population in the region of a study. To extend the agent-based transportation modeling with social influence, a connected synthetic population with both synthetic features and its social networks need to be simulated. However, either the traditional manually-collected household survey data (ACS) or the recent large-scale passively-collected Call Detail Records (CDR) alone lacks features. This work proposes an algorithmic procedure that makes use of both traditional survey data as well as digital records of networking and human behaviors to generate connected synthetic populations. This proposed framework for connected population synthesis is applicable to cities or metropolitan regions where data availability allows for the estimation of the component models. The generated populations coupled with recent advances in graph (social networks) algorithms can be used for testing transportation simulation scenarios with different social factors.

To Mom and Dad

Contents

List of Figures

List of Tables

Acknowledgments

I would like first to express my deepest gratitude to my research advisor Professor Alexei Pozdnoukhov for his help and support ever since I arrived in Berkeley in 2014. Thank you, Alexei, for giving me the opportunity to join your research group. And all the support and encouragement throughout those four years of my graduate study.

I would like to thank Professor Gregory Biging and Professor Joan Walker for serving on my Dissertation Committee, and Professor Scott Moura and Professor Micheal Mahoney for serving on my Qualifying Exam Committee.

I have worked with many amazing individuals in the course of my Ph.D. studies. It has been an honor to do research with Professor Micheal Mahoney, Dr. Kimon Fountoulakis from Department of Statistics and International Computer Science Institute (ICSI), and Junyu Cao from Department of Industrial Engineering and Operations Research. Much of the work done in this dissertation was in collaboration with them, and I owe them immense thanks for all of their encouragement as well as all of the times they pointed out that I was doing something wrong.

I would also like to thank the faculty I interacted with during my time at Berkeley. Many thanks to Laurent El Ghaoui for his Convex Optimization class and Jonathan Shewchuk for his introduction to Machine learning class, where concepts are explained intuitively. I am also grateful that I had the chance to attend the Statistical Learning and Convex Optimization Algorithm classes by Martin Wainwright. I would like to thank Professor Paul Grigas for the insightful discussion on my research. The homework mirror descent with matrix positive semidefinite and trace constraint is still my favorite homework at Berkeley up till now. I thank Professor Steven Glaser and Professor Alexandre Bayen for the approval of my application of switching into the Systems Engineering program. And I still regard my rash decision to switch into the Systems program as the best decision I have ever made. Berkeley also offered me the opportunity to collaborate with amazing students. I have fond memories spending hours deriving algorithms and writing articles with Junyu Cao. I enjoyed discussing with Dounan Tang on diverse topics on economic research for transportation behavior analysis. It was exciting to collaborate with Ye Xia on the attention map prediction project, where I learned a lot about deep learning.

Sid Feygin, Andrew Campbell, Max Gardner, Mogeng Yin, Ziheng Lin, Madeleine Sheehan, Siyu Chen, Yulin Liu, Dr. Yi Liu, Ye Xia, Eric Hou, Sid Masih, Sauhaarda Chowdhuri, Sascha Hornauer, Baladitya Yellapragada, K. Shankari, Professor Joshua Blumenstock, Dr. Jun Ma, Dr. Yun Wang, Dr. Yan Zhang and Dr. Chris Jones, Professor Mark Hansen, Professor Micheal Ball, Professor Paul Waddell, Professor Zuo-Jun Max Shen, Professor David Whitney and Dr. Karl Zipser have been very important collaborators for my research on different projects.

I want to thank all those people who helped me at different phases of my study. Shelley Okimoto, Bernadette Edwards, Helen Bassham and Jeanne Marie have been an immense help during the various phases of the Ph.D. I thank Professor Pingchou Han from College of Engineering, Professor Jian Peng and Professor Guangzhong Cao from College of Urban

and Environmental Sciences for their guidance on my undergraduate research. I want to thank my college roommates Zoe Xu and Yanni Liu, as well as my friend Yanting Shen, for their insightful discussions that attracted me back to science and engineering. I would like thank these Berkeley folks who helped me at the early part of my study at Berkeley: Dounan Tang, Chao Mao, Yi Liu, Tong Wu, Lu Hao, Renyuan Xu, Chaoran Guo, Sid Feygin, Andrew Campbell, Mogeng Yin, Han Cheng, Ziheng Lin, Bowen Yang, Emin Arakelian and Fadi Kfoury. I want to thank all my friends at Berkeley who have seen ups and downs with me: Cheryl Zhang, Jie Ma, Emiko Kawamura, Xining Zang, Jinhan Xie, Marina Yang, Junyu Cao and Flora Zhang. To my teenage friends back home: Ran Huo, Jianing Yang and Yue Yu, thank you for all your encouragement and support.

I want to thank my husband Chen Cheng who offered support and love throughout my Ph.D. We have had a lot of fun memories together, and I am so excited to be sharing my life with you through many more exciting journeys. Last but not least, I need to thank my parents for their support, patience, and love. Thank you, Mom and Dad.

Chapter 1

Introduction

1.1 Motivation

Urban data analytics (or urban computing [140, 139]) is an interdisciplinary field where machine learning/ deep learning methods from computer science and optimization/ simulation methods from operation research are applied in conventional city-related fields, like transportation, environment, social science, ecology, urban planning and civil engineering. Sensing technologies, data management tools and large-scale GPU computing infrastructure have made it much easier to collect, store and analyze big data in urban areas.

For urban data analytics on the topic of transportation machine learning models and transportation simulation, traffic data (loop sensors, surveillance cameras, and GPS in taxis, buses), survey data (ACS, CHTS), mobile phone signals (CDR and GPS) and Location Based Social Network (LBSN) data (Facebook, Twitter, Yelp and Foursquare) have been the main data sources. The availability of these data sources has enabled urban data analytics research at different levels, below are some research topics:

- Marco Traffic Flow Prediction: Call Detail Record (CDR) data and traffic sensor data have been used for route flow (traffic demand) estimation [128, 49, 13] through convex optimization. Loop detectors data and sensor data have been used for traffic forecasting through spatio-temporal machine learning models [28] and deep learning models [73, 18, 29]. While in classic traffic flow forecasting research in transportation engineering and operation research, queuing theory and simulations are the primary methods [33].

- Exploring urban spatial structure: Human mobility data and POIs have been used for identifying functional regions in a city [135, 3]. Taxi trajectories have been used for finding underlying problems of Beijing's road network [136].

- Human Mobility Location Prediction and Activity Recognition: Statistical models like Hidden Markov Chain Model (HMM) [95, 79, 52, 131, 2], Conditional Random Field (CRF) [124] and Input-Output Hidden Markov Chain Model (IO-HMM) model [133] have been used to understand human mobility patterns. IO-HMM [133, 74] has also been used on CDR data to learn individual activity pattern and a corresponding LSTM model [74] has been used to generate activity chain, which is used as the input data to a traffic simulation software MATSim [6].

Social-enabled urban computing adds the third dimension, social, to spatial-temporal urban computing. CDR data and Location Based Social Networks (LBSN) data have been the main data sources for social-enabled urban data analytics research. There has been little research on social-enabled urban data analytics, although urban data analytics is a popular research field. Most of the early works on CDR social network data focus on exploring social networks structure patterns, e.g., the persistence of links with respect to popular node features like degree, centrality, and clustering coefficient [58]. Recent works on networks based on CDR data focus on using location information, and for this most papers focus on large-scale community pattern. For example, in [97] the authors studied how large communities correspond to spatial administrative boundaries, in [42] the authors studied how distance decay effect and spatial continuity control the process of partitioning CDR communities. Other works focus on exploring CDR social networks at the community level, comparing homogeneity of location, age, and mobility pattern within a community, and the heterogeneity of these features between different communities [91, 110]. Furthermore, there has been work on dynamic CDR social network for discovering the relationship between age and size of communities [91]. Other works use CDR data or LBSN data to find individual level pattern, like distance and friendship, mobility and friendship. In [93] the authors found that a large portion of places visited is within several social circles centered at their nearest social ties' locations. In [20] the authors showed that social relationships from CDR data can explain about 10% of all human movement and developed a model using the social network structure to explain periodic short-range movements with travel. LBSN data, like Facebook, Twitter, Yelp, Foursquare, and Gowalla, has mostly been used for point of interests (POI) recommendations based on social network information [17, 75, 137].

1.1.1 Background and motivation for social discrete choice model

Big data is generated every day. It can be generated from online social networks like Facebook and Twitter, web services like Amazon and eBay, and cellular network providers like AT&T and Verizon. A dynamic large-scale social network can be generated based on the big raw data. How to efficiently utilize the large-scale social network information for individual analysis, and how to use small-scale community information to explore collaborative activity patterns of a group of individuals, are interesting topics to explore.

On the other hand, Social influence on human behavior is a topic that both social scientists and computer scientists are interested in, yet they approach this topic in different ways. Social scientists propose various models with latent class and latent variable in the discrete choice modeling framework, and most of the time focus on small graphs. Computer scientists generalize this problem to graph-based problems and propose various methods to deal with large graphs.

Figure 1.1: Large scale graph data

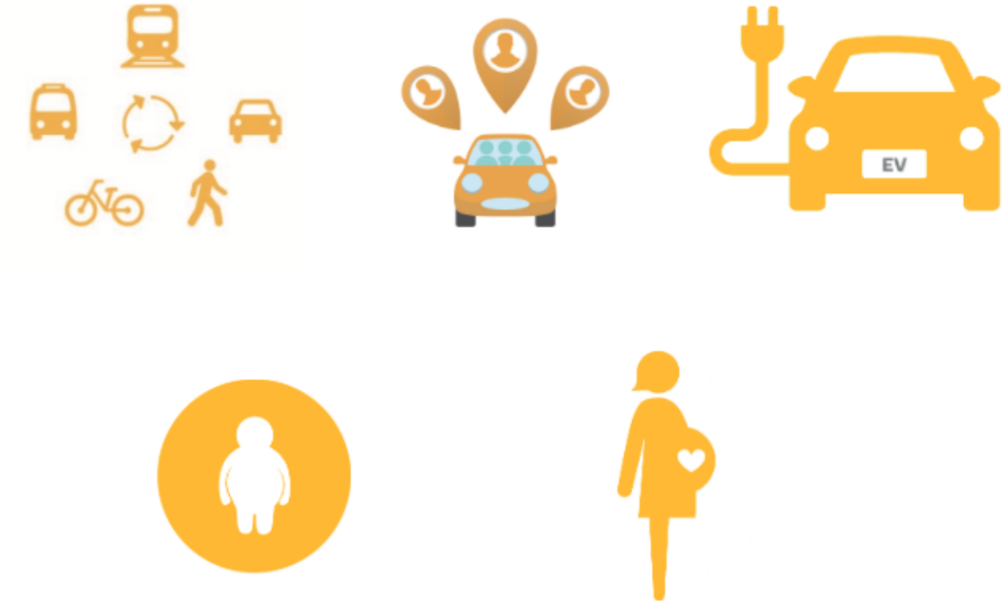


Figure 1.2: Social influence on human behavior

In social science, social preferences have been shown to play an important role in determining the extent to which individuals are likely to exhibit a certain behavior. There is a lot of experimental research in different social science fields that studies the social influence in different ways. For example, (1) recently, travel behavior researchers have become interested in the effects of social influence on travel choice behavior. Conformity behavior as an effect of social influence has been investigated in various qualitative and quantitative studies about travel-mode choice ([34, 94]). There is also some research on the intention to purchase hybrid electric vehicles [5, 63] and attitudes toward bicycling [45]. Recent work studying social influence on car-sharing decision ([64]) shows that people tend to conform to their networks for the car-sharing decision, and the strength of social influence tends to vary according to the social distance. (2) There is also some research studying social influence on health, like obesity [24, 51]and pregnancy [83, 11].

In computer science, the social discrete choice problem is formulated into a semi-supervised graph-based classification problem, for which three mainstream solutions exist. (1) The first category is the graph regularization approach, which adds a graph Laplacian regularization term to the objective function of supervised loss [142, 141, 125]. In this way, connected nodes tend to have similar probability distribution of labels, which means nearby nodes in a graph are likely to have the same labels. (2) The second category is the random-walk based index-context pair approach. This approach introduces the idea of skip-gram framework [81, 82] from natural language processing into node embedding. The objective is to maximize the probability of observing a context based on an index (node), where the context can either be k-hops neighbors of the node [116] or the random path of the node [92, 47]. By adding an extra supervised term to the objective function, both node embedding and classification tasks can be done simultaneously [130]. (3) The third approach is the deep learning approach. Recent development in deep learning has extended the convolutional networks idea from image to graphs [27, 56, 65]. These models are feedforward neural networks that directly apply spectral convolution operations to inputs.

To design a scalable yet predictive model for social discrete choice models is vital for both explanatory research in social science and large-scale machine learning applications in computer science. Our algorithmic models, Local Logistics Graph Regularization (LLGR) and Latent Class Graph Regularization (LCGR), bridge the gap between the two distinct research fields. We also propose Localized Social Discrete Choice Model, combining local graph clustering (LGC) with Local Logistics Graph Regularization (LLGR) for applications on large graphs with local attention.

1.1.2 Background and motivation for social-enabled transportation simulation

Increasing traffic congestion, vehicle emissions and commuters delay have been major challenges for urban transportation systems for years. The economic cost of traffic congestion in the US is increasing from 200 billion in 2013 to 293 billion in 2030 [104]. There is an

Figure 1.3: Challenges for urban transportation system

increasing need for a better solution to long-term transportation demand forecasting for urban infrastructure planning, and solution to short-term traffic prediction for managing existing urban infrastructure. Accordingly, understanding how urban systems operate and evolve through modeling individuals' daily urban activities has been a major focus of transportation planners, urban planners, and geographers. To address operational needs in planning and policy decision making, reliable agent-based land use and urban transportation micro-simulation frameworks such as TRANUS [26], UrbanSIM [118], ILUTE [103], MATSim [6], MEPLAN [23] are becoming popular. Traditional agent-based travel models utilize only the survey data like American Community (ACS) or National Household Travel Survey (NHTS) as the input data. However, the survey data has the problem of data deficiencies and data latencies. Recently, much research has focused on utilizing ubiquitous cellular networks data and location-based services data as input data [133, 60, 62, 74].

However, human aspect in the agent-based travel models is not explored and is an interesting topic. For example, how will your friend influence your commuting trip traveling mode? How will your friends influence your decision to buy an electrical vehicle? What is the difference of the traveling mode between going to the supermarket alone and going to the supermarket together with friends? What is the next activity if I go to the supermarket alone or go to the supermarket with my friends? If I am carpooling with someone, when will I prefer to carpool with him again or hang out with him? How to design a socially and environmentally car-pool mechanism for commuting trips?

To answer these questions, we have to enable the current agent-based travel models with social information. To efficiently incorporate social information into the agent-based modeling framework, below three parts need to be modified from Figure 1.4, as shown in Figure 2.1: (1) connected synthetic population, with detailed information on each of the agents and the social connections between individuals, is needed to be provided as input to the micro-simulation model; (2) activity plan generator with social interaction, is needed to generate the activity plans for the connected synthetic population, which is also the input to the micro-simulation model; (3) social discrete choice model is needed to model the route choice of individuals given activity plans and social networks; (4) social carpool scenarios mechanism design is needed to incorporate the social carpooling mode choice into the microsimulation system.

The review of the state-of-the-art models in different components are reviewed separately in different chapters of this thesis. Chapter 3 and chapter 5 are self-contained research papers.

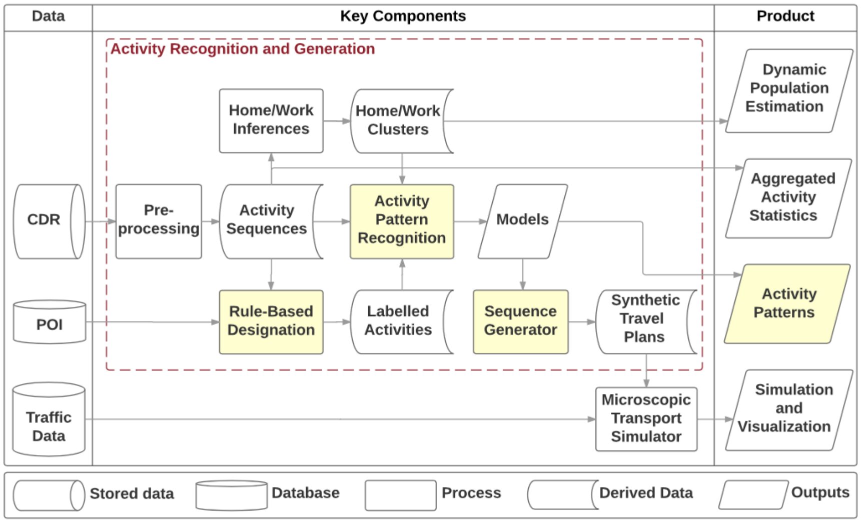


Figure 1.4: Agent-based travel model utilizing CDR and ACS [133]

Chapter 2

Problem Statement and Contributions

2.1 Objective and challenges

This dissertation aims to explore two directions for future research on social-enabled urban data analytics:

- We explored how to develop new machine learning models well suited to the social urban data analytics research. We focus on modeling social influence on human behavior from a graph modeling perspective while conforming to the discrete choice modeling framework.

- We explored how to incorporate social factor into the existing urban simulation framework of urban data analytics. We focus on developing an algorithmic procedure that makes use of both traditional survey data as well as digital records of networking and human behaviors to generate connected synthetic populations.

This dissertation also provides the derivation and formulation of all required steps of the Alternating Direction Method of Multipliers (ADMM) algorithm in the context of the specific type of graph regularization with both global and local variables, as well as the Expectation Maximization (EM) algorithm in the context of the latent class model and the latent class graph regularization model. A localized version of the proposed social discrete choice model, which can deal with huge graph for certain type of modeling task, is also introduced in this dissertation. Throughout this dissertation, models are tested on real-world small data and big data. Predicting and modeling results are presented to empirically demonstrate the performance of our models.

2.2 Proposed Research Pipeline

This thesis explores how to add social information to urban data analytics, with the potential application in social-enabled transportation simulation. The thesis is organized in the below

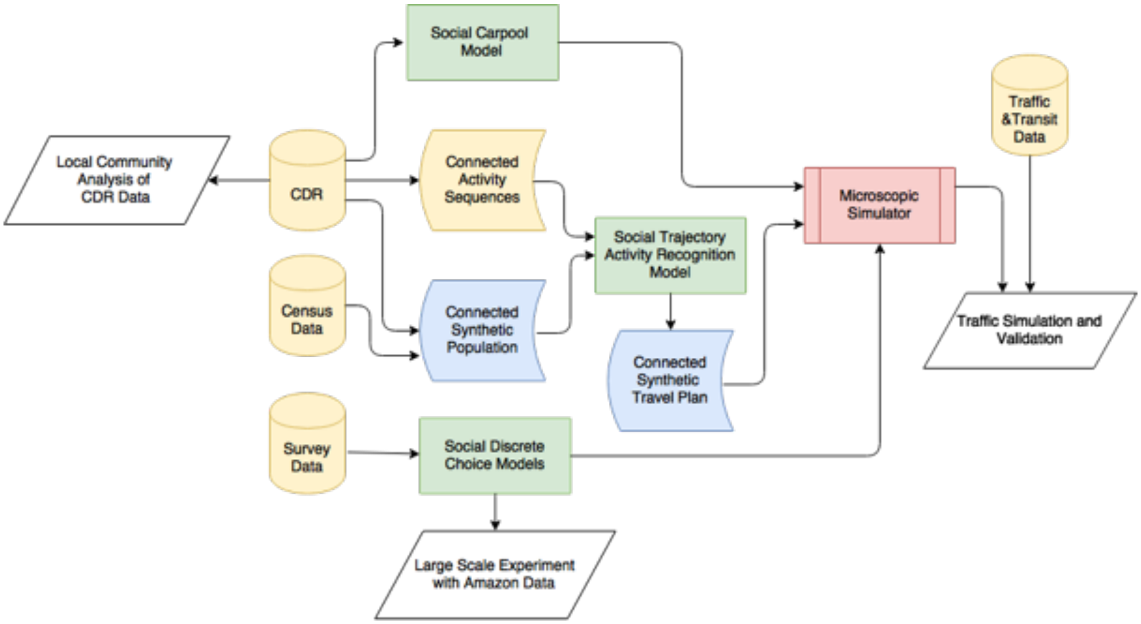

Social Carpool Model
Local Community Analysis of CDR Data
CDR
Connected Activity Sequences
Traffic &Transit Data
Microscopic Simulator
Social Trajectory Activity Recognition Model
Census Data
Connected Synthetic Population
Connected Synthetic Travel Plan
Traffic Simulation and Validation
Survey Data
Social Discrete Choice Models
Large Scale Experiment with Amazon Data

Figure 2.1: Proposed research pipeline

way

- **chapter3** The central topic of social-enabled transportation simulation is how friends influence individual's travel mode choice and other transportation related choices. In chapter 3, two novel machine learning algorithms: Local Logistics Graph Regularization (LLGR) and Latent Class Graph Regularization (LCGR) are proposed to solve the social discrete choice models as a graph-based semi-supervised classification problem. Both small scale and large scale experiments are conducted to illustrate the usefulness of the methodology.

- **chapter4**. Chapter 4 entails several parts: (1) the discussions of the social discrete models; (2) the derivation and formulation of latent class model, LLGR and LCGR models: detailed line by line derivation and proof of correctness are provided; (3) extension of social discrete choice models: localized social discrete choice model.

- **chapter5** In chapter 5, a novel data simulation framework is proposed to make use of both traditional survey data as well as digital records of networking and human behaviors to generate connected synthetic populations. The generated populations coupled with recent advances in graph (social networks) algorithms can be used for testing transportation simulation scenarios with different social factors.

- **chapter6** Chapter 6 provides a comprehensive summary of the research motivation, objective, adopted methodological frameworks and corresponding findings. This chapter also focuses on identifying future research directions for the proposed social-enabled urban data analytics.

List Of Abbreviations Used

ADMM	Alternating Direction Method of Multipliers
LSTM	Long-Short Term Memory
GCN	Graph Convolutional Networks
LLGR	Local Logistics Graph Regularization
LCGR	Latent Class Graph Regularization
LTD-Graph LSTM	Localized Temporal Dynamic Graph LSTM
CDR	Call Detail Records
ACS	American Community Survey
NHTS	National Household Travel Survey
TAZ	Traffic Analysis Zone
LGC	Local Graph Clustering
DCM	Discrete Choice Models
EM	Expectation Maximization Algorithm
MCEM	Monte Carlo Expectation Maximization Algorithm
NLP	Natural Language Processing
word2vec	word2vec word embedding algorithm from NLP
node2vec	node2vec node embedding algorithm
MCMC	Markov chain Monte Carlo algorithm
ERGM	Exponential Random Graph Model
IPF	Iterative Proportional Fitting
BIC	Bayesian information criterion
AIC	Akaike information criterion
LLBP	Lagrangian Relaxation Lower Bound
MO	Modus Operandi
DNN	Deep Neural Networks
NCP	Network Community Profile
DCP	Distance Community Profile
DnCP	Density Community Profile
LBSN	Location Based Social Networks
HMM	Hidden Markov Model
IOHMM	Input- Output Hidden Markov Model
RNN	Recurrent Neural Networks
LSTM	Long short-term memory

Table 2.1: Abbreviations

2.3 Key Contributions

The key contributions of this research may be summarized as:

- Social Discrete Choice Models (chapter3)
  - Proposed the first model that combines graph regularization with latent class model, with rigorous mathematical formulation
  - Proposed scalable parameter estimation strategy for both LLGR and LCGR models.
  - Empirically illustrated the impact of graph structure by varying the connectivity between different classes in the graph, and illustrated the impact of label assignment by varying the discrepancies of labels in communities of the graph.
  - Analyzed relative performance of the Local Logistics Graph Regularization (LLGR) and Latent Class Graph Regularization (LCGR) models compared with baseline logistics regression model and latent class model, on small scale dataset
  - Analyzed relative performance of the Local Logistics Graph Regularization (LLGR) model compared with baseline logistics regression model, Graph Convolutional Networks (GCN) model, random-walk index-context pair node embedding model on large scale dataset
- Social Discrete Choice Models Appendix
  - Provided empirical result of parallel block Markov chain Monte Carlo (MCMC) and explored the possibilities of running this algorithm in E step fo the Latent Class Graph Regularization (LCGR) model.
  - Provided the derivation and formulation of all required steps of the Alternating Direction Method of Multipliers (ADMM) algorithm in the context of the specific type of graph regularization with both global and local variables, as well as the Expectation Maximization (EM) algorithm in the context of the latent class model and the latent class graph regularization model.
  - Proposed Localized Social Discrete Choice Models, which benefits of both Social Discrete Choice Model and Local Graph Clustering. The model is tested on a real world online retail account relation graph from a leading commercial company in the US, for the task of graph-based fraud detection.
- Connected Population Synthesis (chapter5)
  - Proposed the first algorithmic procedure that makes use of both traditional survey data as well as digital records of networking and human behaviours to generate connected synthetic populations.
  - The generated synthetic population can replicate the below properties
    - marginal and joint distributions of individuals and household level socio-economic characteristics as the American Community Survey (ACS)

- geographical pattern of the observed community structure as from the Call Detail Record (CDR)
- the graph statistics of the observed Call Detail Record (CDR) social networks.

Chapter 3

Social Discrete Choice Models

3.1 Introduction

In this paper, we focus on how to efficiently incorporate social network information into latent class models for user discrete choice modeling problems. Traditional models ignore social information and make the assumption that labels are separable in the feature space. However, for many life-style related choices (such as bicycling vs. driving to work, smoking vs. not smoking, overeating vs. not overeating), social considerations are thought to be a key factor. People with very similar characteristics can have very different choices, often thought to be due to the influence of their friends. Also, people who make similar lifestyle related choices are thought to be more likely to be connected and form communities. Although these "birds of a feather flock together" phenomena are widely studied in social sciences, no existing predictive model can efficiently solve this problem in computational social science. In this paper, we reformulate the problem from discrete choice settings into a graph-based semi-supervised classification problem.

We propose two models to efficiently exploit the social network (graph) information. (1) The first model is the local logistics graph regularization (LLGR) method. Parameter estimation of this model is performed using a specialized Alternating Direction Method of Multipliers (ADMM), where the computation of each node can be parallelized, making the algorithm very scalable to large graphs. (2) The second model is the latent class graph regularization (LCGR) model, where we aim to combine the expressiveness of parametric model specifications with descriptive exploratory power of latent class models. Parameter estimation of the LCGR model is performed using a specialized Monte Carlo expectation maximization algorithm presented in Section 3.5. We adopt the same ADMM techniques for the M step and discuss the parallel computation for the E step in Section 4.1. The LCGR model can outperform the LLGR model, but it is computationally more expensive. We recommend using the LCGR model for small graphs and the LLGR model for large graphs (both of which are of interest in web applications).

To illustrate the usefulness of our methodology, we look at three classes of data. (1) The first class is small synthetic data used to illustrate how the knobs of our methods perform in idealized and less-than-idealized situations. We experimented with our methods by tuning the class connectivity hyperparameter β and choice preference hyperparameter w. When labels are not separable in feature space (which means linear hyperplanes that separate the data x_i accurately do not exist), but are separable in the graph space (which means decisions y_i are clustered based on communities in the graph), our model outperformed all other baseline models. In other cases, our model performed no worse than other models. (2) The second class is small real data used to illustrate how our method performs on a typical example of interest to social scientists, and we compared with the state-of-arts methods in social science. We experimented with our models on real-world adolescent smoking dataset from 1995 to 1997. We found out that the smoking preferences were largely defined by the objective factors for those adolescents at first in 1995. But smoking within a certain group of teenagers became a social norm in 1997, and our social discrete choice model performed much better than models that ignore social networks structure. (3) The third class is a large-scale example from internet analysis used to illustrate how our method can be expected to perform in the larger-scale internet and social media applications, compared with other scalable methods [47, 27, 65]. A large-scale experiment is conducted on an online retail account relation graph for fraud detection. Our method is more robust than other semi-supervised graph-based classification methods on a graph with huge components and high average degree, which is very common in real-world applications.

3.2 Problem Formulation

Social discrete choice model is to utilize social network data with discrete choice models, to study social influence on human behavior. We illustrate how social discrete choice model is formulated by an example: adoption of electric vehicles (Figure 3.1).

As shown in Figure 3.1, each individual is represented by a vertex in the graph, we connect two vertices by an edge if they are friends with each other. Apart from the social network information, individual's socio-economic and demographic information is also given as feature vector. The color of node represents whether an individual is willing to buy an electric vehicle: "red" represents not willing, "green" represents willing. As we can observe from the graph, only part of the nodes have color, which means we only know the choices of some of the individuals. And we are interested in predict the choice of the rest of the individuals, which are represented by white nodes in the graph. So the social discrete choice model is to predict the choices the rest of the people in the graph, given the graph structure and the choices of some of the individuals as well as the feature vector of everyone.

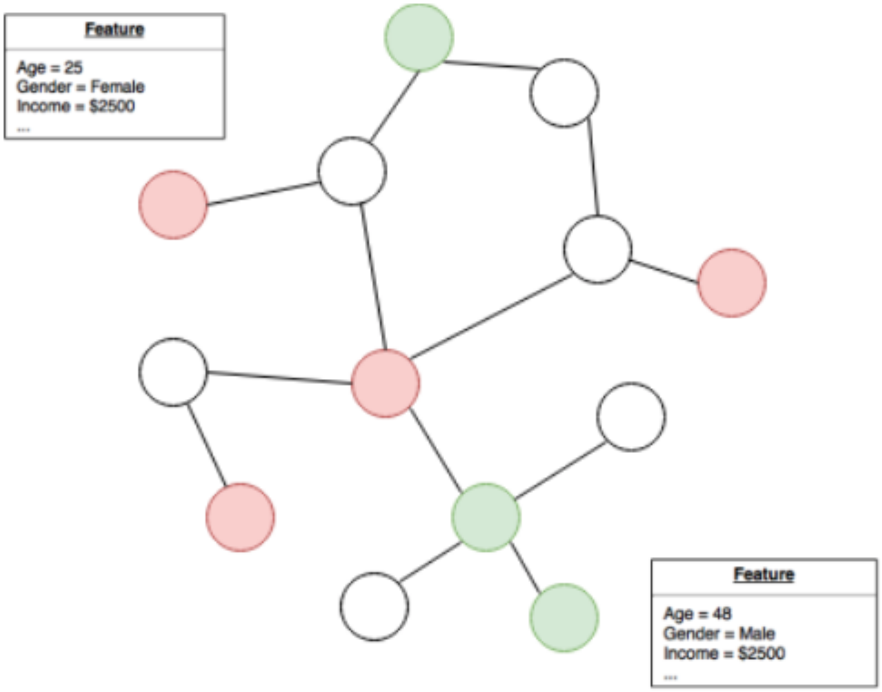


Figure 3.1: Problem formulation of social discrete choice model: the study of social influence on the adoption of electric vehicles. Each individual is represented by a vertex in the graph, we connect two vertices by an edge if they are friends with each other. The color of a node represents whether an individual is willing to buy an electric vehicle: “red” represents not willing, “green” represents willing.

3.3 Literature Review

Social discrete choice model is a topic that both social scientists and computer scientists are interested in, yet they approach this topic in different ways. Social scientists propose various models with latent class, and most of the time focus on small graphs. Computer scientists generalize this problem to graph-based semi-supervised classification and come up with various of methods to deal with large graphs.

3.3.1 Discrete Choice Modeling and Latent Class Model

In many application domains, human decision making is modeled by discrete choice models. These models specify the probability that a person chooses a particular alternative from a given choice set, with the probability expressed as a function of observed and unobserved latent variables that relate to the attributes of the alternatives and the characteristics of the person. Multinomial logit models are in the mainstream of discrete choice models, with maximum likelihood used for parameter estimation from manually collected empirical data. It is important for practitioners to interpret the observed choice behaviors, and models that are linear in parameters are most common. At the same time, choice preferences within different social groups (though seemingly similar in terms of the observed characteristics) can vary significantly due to the unobserved factors or different context of the choice process. One way of accounting for this is to introduce latent class models. Latent class logistic regression models are common tools in multiple domains of social science research [22, 101, 120].

It is also recognized that social influence can be a strong factor behind variability in choice behaviors. The impact of social influence on individual decision-making has attracted a lot of attention. Researchers have employed laboratory experiments, surveys, and studied historical datasets to evaluate the impact of social influence on individual decision making. However, it is difficult to avoid an identification problem in the analysis of influence processes in social networks [78]. One has to account for endogeneity in explanatory variables in order for claims of causality made by these experiments to be useful [35]. Due to these limitations of observational studies of influence, randomized controlled trials are becoming more common. In general, distinguishing social influence in decision making from homophily, which is defined as the tendency for individuals with similar characteristics and choice behaviors to form clusters in social networks, is currently a growing area of research and debate [109, 21].

3.3.2 Graph-based semi-supervised classification

Our graphical extension to the latent class model reformulates the social discrete choice problem into a semi-supervised graph-based classification problem, for which three mainstream solutions exist.

3.3.2.1 Graph Regularization approach

Graph regularization methods that penalize parameter differences among the connected nodes have been studied in the context of classification, clustering, and recommendation systems [1, 76]. Graph-based semi-supervised learning of this kind adds a graph Laplacian regularization term to the objective function of supervised loss [142, 141, 125, 8]. The graph Laplacian regularization term in the loss function is shown as below, where Δ is the graph Laplacian matrix:

$$\lambda \sum_{(i,j)\in\mathcal{E}} a_{i,j}||f(x_i) - f(x_j)||^2 = \lambda f^T(A-D)f = \lambda f^T \Delta f \tag{3.1}$$

where A is the adjacency matrix, D is the diagonal matrix, and f is the function that maps feature vector into probability distribution vector.

These models assume that connected nodes tend to have similar probability distribution of labels, which means nearby nodes in a graph are likely to have the same labels. For parameter estimation, Zhu et al. [142] and Zhou et al. [141] proposed diffusion-based learning algorithms that involve solving linear systems directly using matrix operations. Gleich et al.[43] reformulated the diffusion-based learning problem into an optimization problem and added l1 regularization term to obtain a more robust solution. And a local push algorithm as in [4] was introduced to calculate a local solution.

On the other hand, our proposed semi-supervised latent class classification has simpler problem formulation. Our model assumes that connected nodes tend to have similar local classifier. That means connected nodes have similar probability distribution of labels only when they have similar feature vectors. Our graph regularization objective function is shown below, where notations are defined in Table 5.1:

$$\min \sum_{i \in \mathcal{V}} \log \left(1 + e^{-y_i \times (W^T x_i + b_i)}\right) + \lambda \sum_{(i,j) \in \mathcal{E}} (b_i - b_j)^2 \tag{3.2}$$

In addition, we used the ADMM method to speed up the algorithm by distributing the computation, owing to recent advances in distributed optimization applied to parametric models on networks [50].

3.3.2.2 Random-walk based index-context pair approach

The recently developed skip-gram model is widely used in learning word embedding [81, 82] and node embedding, both in unsupervised and semi-supervised manner. The objective is to maximize the probability of observing a context based on an index (node), where the context can either be k-hops neighbors of the node [116] or the random path of the node [92, 47]. By adding an extra supervised term to the objective function, both node embedding and classification tasks can be done simultaneously [130].

3.3.2.3 Deep Learning Approach

Recent development in deep learning has extended the convolutional networks idea from image to graphs [27, 56, 65]. These models are feedforward neural networks that directly apply spectral convolution operations to inputs. Henff et al. [27] used K-localized convolution to replace the spectral convolution operations, Kipdf et al. [65] used a linear model as a first-order approximation of localized spectral filters. Graph Convolutional Networks (GCN) [65] is computationally less expensive than CNN on graphs [27], and the authors claimed the model can outperform all other models on the public dataset for semi-supervised classification problem.

3.4 Social Models

3.4.1 Notations

We define $[N] := \{1, 2, \ldots, N\}$, $i \in [N]$, $t \in [K]$, where N and K are integers. We will use the following notations and definitions. For simplicity, in this paper, we focus on binary discrete choice case, which can be easily extended to multinomial discrete choice.

We assume that there is only one sample per node, i.e., $n = 1$. However, the proposed models can be extended to the case of $n > 1$. We further assume that the graph $(\mathcal{V}, \mathcal{E})$ is

Table 3.1: Notation: Table for notations

Variable	Definition
N	number of individuals
K	number of latent classes
n	number of samples per node
d	number of features for each individual
$x_i \in \mathbb{R}^{d \times n}$	feature-samples matrix of individual i
$z_i \in [K]$	latent class variable of individual i
$y_i \in \{-1, 1\}$	binary choice of individual i
$W_t \in \mathbb{R}^d$	model coefficients of class t
$b_{it} \in \mathbb{R}$	model offset coefficients of individual i with class t
$\mathcal{V}$	set of nodes in a social graph, with each node corresponding to an individual
$\mathcal{E}$[1]	set of edges, presenting relationship between two individuals

unweighted, noting that the models can be extended to weighted graphs. In the following subsections we occasionally drop indices i and t depending on the context to simplify notation. We denote with $\theta := \{W, b\}$ the set of model coefficients W_t, b_{it}, $\forall i, t$.

Let $h_{it}(x_i) := W_t^T x_i + b_{it}$, we consider the probability distribution for the choice of individual i in class t, as below:

$$P(Y_{it} = y_{it}) = \frac{1}{1 + e^{-y_{it} h_{it}(x_i)}} \tag{3.3}$$

where y_{it} can take values of 1 or -1. Note that the following two state-of-the-art discrete choice models follow this probability distribution: (1) logit discrete choice model without alternative specific attributes, which is proved to be equivalent to the logistics regression model [84], where $t \equiv 1$ and $b_{it} = b_{jt}, \forall \{i, j\} \in \mathcal{V}^2$; (2) latent class model, where $t > 1$ and $b_{it} = b_{jt}, \forall \{i, j\} \in \mathcal{V}^2, \forall t$. We also define our models according to the probability distribution in Equation (3.3): (1) LLGR model where $t \equiv 1$, and b_{it} are not constant; (2) LCGR model where $t \equiv 1$, and b_{it} are not constant.

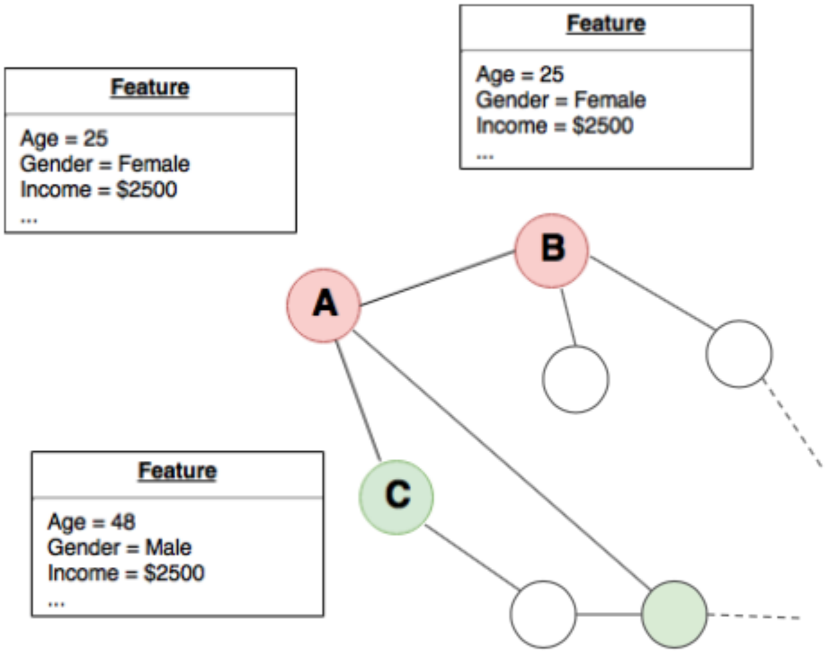


Figure 3.2: Adoption of electric vehicles example. Each individual is represented by a vertex in the graph, we connect two vertices by an edge if they are friends with each other. The color of node represents whether an individual is willing to buy electric vehicle: "red" represents not willing, "green" represents willing.

3.4.2 Motivation for the first model

Figure 3.2 is a snapshot of the large social network for the study of adoption of electric vehicles. As defined in Figure 1.1, the color of a node represents whether an individual is willing to buy an electric vehicle: "red" represents not willing, "green" represents willing. As shown in the graph, node A and node B, node A and node C are connected. Node A and node B have exactly the same feature vectors, while node A and node C have different feature vectors. Node A and node B are not willing to buy electric vehicles, while node C is willing to buy electric vehicles. The old graph regularization model [142, 141, 125, 8] assumes connected nodes are more likely to have similar probability distribution of labels no matter what their feature vectors are, because the optimization objective function as in equation 3.4 are forcing $f(x_i)$ and $f(x_j)$ equal to each other when node i and node j are connected, and $f(x_i)$ is the probability distribution or its variants for node i.

$$\lambda \sum_{(i,j)\in\mathcal{E}} a_{i,j}||f(x_i) - f(x_j)||^2 = \lambda f^T(A-D)f = \lambda f^T \Delta f \tag{3.4}$$

However, based on the observation from Figure 3.2 and everyday life, feature vectors indeed have influence on choices for connected individuals. Although node A and node B are connected, and node A and node C are also connected. Node A and node B have the same choice because they have the same input feature vector, and node A and node C have different choices because they have different input feature vectors. This is the motivation for the local logistic graph regularization (LLGR) model, where the underlying statistical assumption is that only when connected nodes have similar feature vectors that they are more likely to have similar probability distribution of labels.

3.4.3 Local logistic graph regularization (LLGR)

Canonical graph regularization models [142, 141, 125, 7] assume connected nodes are more likely to have similar probability distribution of labels. However, friends can make quite different choices because they have different features such as age, gender, and income. Our LLGR model also emphasizes the importance of individual feature vector. The LLGR model assumes that only if two nodes are connected and have similar feature vectors, they are likely to have similar probability distribution of labels.

The LLGR model is also included in the choice model specified by Eq. (3.3). In the LLGR model there is no latent class, so t is removed, $K = 1$ and y_i follows a Bernoulli distribution given x_i. To incorporate the social aspect in logistic regression one assumes that the parameters b follow an exponential family parameterized with the given graph

$$P(b) \propto \prod_{(i,j)\in\mathcal{E}} e^{-\lambda(b_i-b_j)^2}, \tag{3.5}$$

where $\lambda \in \mathbb{R}$ is a hyper-parameter. This model is usually trained by using maximum a posterior (MAP) estimator which reduces to the following regularized logistic regression problem

$$\theta^* := \operatorname*{argmin}_{\theta} \sum_{i=1}^{N} \log\left(1 + e^{-y_i h_i(x_i)}\right) + \lambda \sum_{(i,j)\in\mathcal{E}} (b_i - b_j)^2, \tag{3.6}$$

where $h_i(x_i) := W^T x_i + b_i$. Notice that the social information, i.e., edges $\mathcal{E}$, appears as Laplacian regularization for the coefficients b.

3.4.4 Motivation for the second model

Figure 3.3 is another snapshot of the large social network for the study of the adoption of electric vehicles. The connections and feature vectors of nodes A, B and C are the same as in figure 3.2. Node A and node B have different choices, although node A and node B are connected and have the same feature input vector. If this phenomenon is observable across the graph, then the observation is contradictory to the statistical assumptions of the LLGR model. Inspired by the latent class model, the natural explanation is that node A and node B

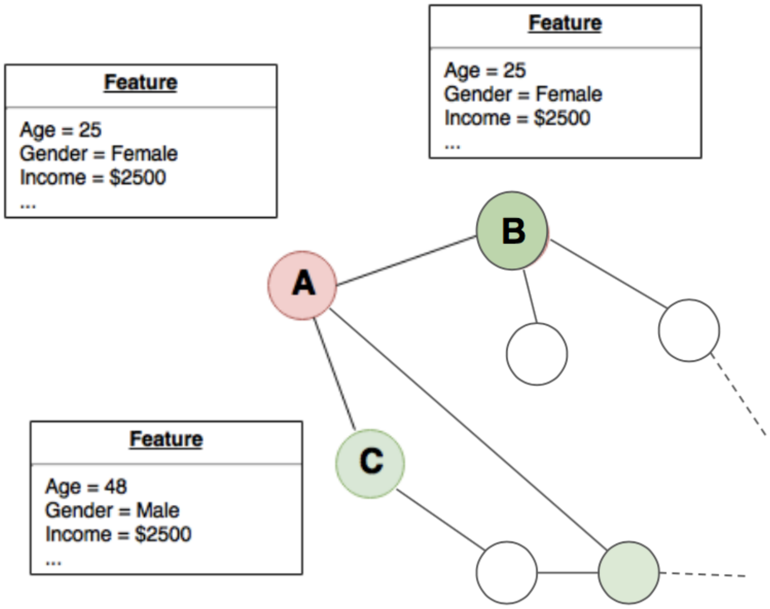


Figure 3.3: Adoption of electric vehicles example. Each individual is represented by a vertex in the graph, we connect two vertices by an edge if they are friends with each other. The color of a node represents whether an individual is willing to buy electric vehicle: "red" represents not willing, "green" represents willing.

have some unobservable heterogeneity that they are in different class. This is the motivation for the Latent class graph regularization (LCGR) model, where graph regularization model interacts with the latent class model.

3.4.5 Latent class graph regularization (LCGR)

However, connected nodes with similar feature vectors not always have similar probability distribution because unobserved heterogeneity among individuals exists such as taste differences. To solve this issue, we propose the LCGR model to combine social relations and latent class. It is an extension to LLGR model with latent classes where $K > 1$. In this model, y_{it} follows a Bernoulli distribution given x_i and $z_i = t$. To incorporate social information, we assume that latent class variables z_i are distributed based on the following exponential family parametrized by the given social graph

$$P(z; b) \propto \prod_{(i,j)\in\mathcal{E}} \exp\left(-\lambda \sum_{t=1}^{K} (b_{it} - b_{jt})^2 \mathbf{1}(z_i = z_j = t)\right), \tag{3.7}$$

where b represents the collection of coefficients b_{it} $\forall i, t$, which are the parameters of the distribution.

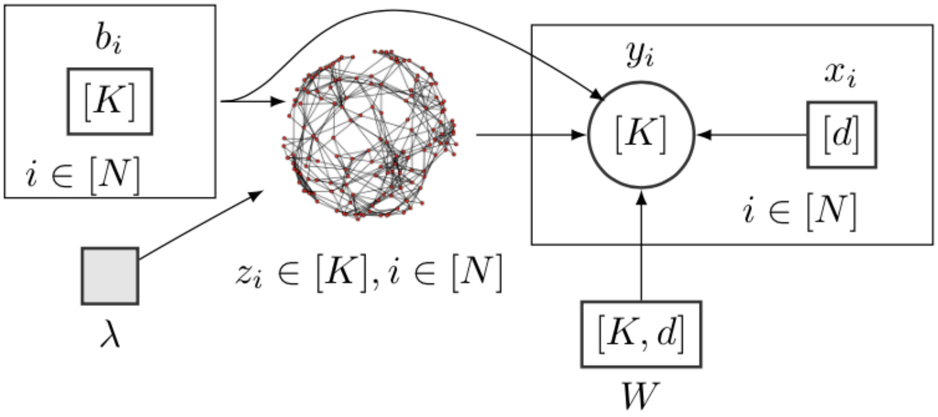


Figure 3.4: Graphical model representation for social logistic regression models with latent variables. We use a modified plate notation to represent conditional dependence among random variables and dependence on parameters. In particular, random variables are represented using circles and their number is shown in brackets inside the circle, i.e., y_i corresponds to K variables. Parameters are represented in rectangles, and their sizes are shown in brackets with two components, i.e., W corresponds to $K \times d$ coefficients. Data are shown in rectangles and their size in brackets, i.e., x_i corresponds to d features. There are N nodes in the graph and each node corresponds to a random variable z_i which takes values in $[K] := \{1, \ldots, K\}$. The hyper-parameter λ is represented using a grey rectangle.

Note that this specification allows utilizing social structures by introducing (1) continuous latent variables b_{it} defined in graph regularization; (2) discrete latent variables z_i defined in the above Markov Random Field. For continuous latent variables b_{it} in this model, we assume that each individual has its own local coefficient b_{it} for each class. Notice that this model does not penalize different coefficients b_{it} among connected individuals in different classes. This is because we assume that connected individuals in different classes should have independent linear classifiers. For discrete latent variables z_i in the common latent class models, they are independent and identically distributed following the multinomial distribution, i.e., $z_i \sim \text{Mult}(\pi, 1)\ \forall i$, where π is the probability of success. However, in our specification, hidden variables z are correlated and are not necessarily identically distributed. Hence the continuous latent variables b and the discrete latent class variable z in our model can better model the observed choice processes, through which we can improve the model performance compared with the state-of-the-art models.

A graphical interpretation of this model is given in Figure 3.4. The resulting model can be trained using maximum likelihood and the Expectation-Maximization (EM) algorithm; details are discussed in Section 3.5.

3.5 Parameter Estimation

In this section, we focus on the parameter estimation algorithms for both the LLGR model and the LCGR model.

3.5.1 Local logistics graph regularization: ADMM

In this section, we discuss how to conduct optimization in a distributed manner for maximum a posterior probability (MAP) parameter estimation. Following the work of [50] that applies ADMM to network lasso method, we extend it by allowing both local b and global variables W on nodes. Let

$$Q(\theta; x, y) := \sum_{i\in\mathcal{V}} \log\left(1 + e^{-y_i h_i(x_i)}\right) + \lambda \sum_{(i,j)\in\mathcal{E}} (b_i - b_j)^2 \tag{3.8}$$

be the objective function, where θ represents the collection of parameters W and b and $h_i(x_i) := W^T x_i + b_i$. To minimize (3.8) using ADMM, we introduce a copy of b_i denoted by s_{ij}, $\forall i$, and a copy of W denoted by g_i, $\forall i$,

$$\begin{array}{ll} \min\limits_{W,b} & \sum\limits_{i\in\mathcal{V}} \log\left(1 + e^{-y_i(g_i^T x_i + b_i)}\right) + \lambda \sum\limits_{(i,j)\in\mathcal{E}} (s_{ij} - s_{ji})^2 \\ \text{s.t:.} & b_i = s_{ij} \;\; j \in \mathcal{N}(i), \;\; \forall i \\ & W = g_i, \;\; \forall i, \end{array} \tag{3.9}$$

where $\mathcal{N}(i)$ are the adjacent nodes of node i. By introducing copies for b_i, $\forall i$, we dismantle the sum over edges into separable functions. Additionally, by introducing copies for W, we dismantle the sum over the nodes for the logistic function. Then by relaxing the constraints we can make the problem (3.9) separable, which allows for distributed computation.

We define the augmented Lagrangian below, where u and r are the dual variables and ρ_1 and ρ_2 are the penalty parameters.

$$\begin{aligned} L_{\rho,\lambda}(W, b, g, s, u, r) := \sum_{i\in\mathcal{V}} \Big\{ \log\left(1 + e^{-y_i(g_i^T x_i + b_i)}\right) \\ + \frac{\rho_1}{2}\Big(-\|r_i\|_2^2 + \|W - g_i + r_i\|_2^2 \Big)\Big\} + \sum_{(i,j)\in\mathcal{E}} \Big\{ \lambda (s_{ij} - s_{ji})^2 \\ + \frac{\rho_2}{2}\Big(-\|u_{ij}\|_2^2 - \|u_{ji}\|_2^2 + \|b_i - s_{ij} + u_{ij}\|_2^2 + \|b_j - s_{ji} + u_{ji}\|_2^2 \Big)\Big\} \end{aligned}$$

The resulting ADMM algorithm is presented in Algorithm 3, where

$$f(s_{ij}, s_{ji}) := L_\rho(W^{k+1}, b^{k+1}, g^{k+1}, (s_{ij}, s_{ji}, s^k_{(ij)^c}), u^k, r^k)$$

Notice that the subproblems in Step 4 do not have closed form solutions. However, they can be solved efficiently using an iterative algorithm since they are univariate problems that depend only on x_i and not all data. Similarly, the subproblems in Step 5 do not have closed form solution, but they have only d unknown variables and depend only on x_i and not all data. Moreover, Step 6 has a closed form solution, which corresponds to solving a 2×2 linear system. Observe that the ADMM algorithm 3 can be run in a distributed setting by distributing the data among processors, because within each iteration, the computation of the value update of each node and edge are independent. Figure 3.5 demonstrates how distributing the data among processors can speed up the convergence of the ADMM algorithm. In this experiment, we randomly generate a binomial graph with 100k nodes, 500k edges. Then we randomly generate the feature matrix and response vector for the graph. We test the model on a server with 12 processors, and we use the multiprocessing package from Python to control the number of processors used in the parallel computing paradigm. As can be seen from Figure 3.5, distributing the computation among processors can greatly reduce the running time. Note that the running time does not decrease proportionally to the number of processors. It is because Python multiprocessing module is used here, but it takes time for process to communicate with the memory, and only the step for updating b_{it} is parallelled.

Algorithm 1 ADMM for Problem 3.9

```
1: **Initialize:**
       $k \leftarrow 0$, $W^k$, $b^k$, $g^k$, $s^k$, $u^k$ and $r^k$
2: **repeat**
3:     Set $W_t^{k+1} = \frac{1}{N}\sum_{i=1}^{N}(g_i^k - r_i^k)$
4:     $b_{it}^{k+1} := \arg\min_{b_i} \; L_\rho(W_t^{k+1}, b_i, g^k, s^k, u^k, r^k) \; \forall i \in \mathcal{V}$
5:     $g_i^{k+1} := \arg\min_{g_i} \; L_\rho(W^{k+1}, b^{k+1}, g_i, s^k, u^k, r^k) \; \forall i \in \mathcal{V}$
6:     $s_{ij}^{k+1}, s_{ji}^{k+1} = \arg\min_{s_{ij}, s_{ji}} \; f(s_{ij}, s_{ji}) \; \forall (i,j) \in \mathcal{E}$
7:     Set
           $r_i^{k+1} = r_i^k + \rho_1(W^{k+1} - g_i^{k+1}) \; \forall i \in \mathcal{V}$
           $u_{ij}^{k+1} = u_{ij}^k + \rho_2(b_i^{k+1} - s_{ij}^{k+1}) \; \forall (i,j) \in \mathcal{E}$
           $u_{ji}^{k+1} = u_{ji}^k + \rho_2(b_j^{k+1} - s_{ji}^{k+1})$
8:     $k \leftarrow k + 1$
9: **until** termination criteria are satisfied.
```

3.5.2 Latent Class Graph Regularization: Monte Carlo EM

Generally, graphical models with categorical latent variables can be solved using the expectation maximization (EM) algorithms. However, correlations among latent variables imposed by the social graph do not allow exact calculation of posterior distributions in the E-step using

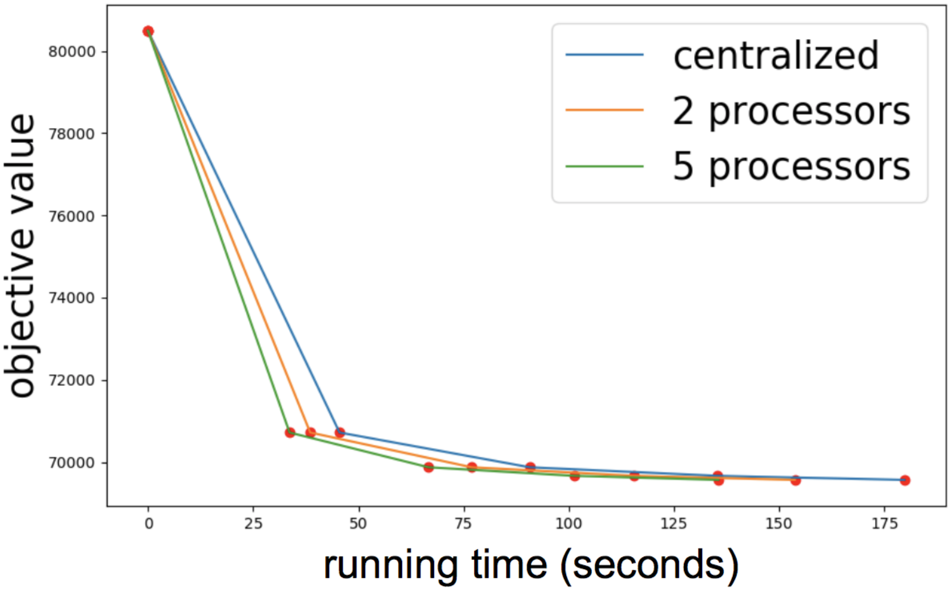


Figure 3.5: Illustration of the performance of ADMM algorithm with different number of processors. Each node represents the running time and objective value of a iteration in an experiment

standard EM approaches. Instead, an approximate calculation of the E-step using Monte Carlo EM (MCEM) [86, 72, 30, 123] is employed. It is a modification of the original EM algorithm where the E-step is conducted approximately using a Monte Carlo Markov Chain (MCMC) algorithm. The details for each step of the MCEM algorithm for the proposed models are provided in the following subsections.

3.5.2.1 Expectation step

In this step the objective is to compute the marginal posterior distribution for nodes and edges, which will be used in the M-step to calculate the negative expected log-likelihood function. Refer to the Appendix for derivation of negative expected log-likelihood, which reveals the need for calculation of marginal posterior distributions.

In particular, for the E-step, one needs to calculate the following node marginal posterior probability

$$P(z_i = t|y_i, x_i; \theta) = \frac{P(y_i|x_i, z_i = t; \theta)P(z_i = t; b)}{\sum_{s=1}^{K} P(y_i|x_i, z_i = s; \theta)P(z_i = s; b)} \tag{3.10}$$

Algorithm 2 MCEM algorithm for LCGR

```
Inputs:
    $(x_i, y_i)$, $i = 1 \dots N$
Initialize:
    $\theta^0 := \{W^0, b^0\} \leftarrow$ arbitrary value, $k \leftarrow 0$
repeat
    E-step: (Subsection 3.5.2.1)
    Calculate approximate node posterior
for each node $i \in [N]$
        $q(z_i = t) := P(z_i = t | y_i, x_i; b^k)$
,     and for each edge $(i, j) \in \mathcal{E}$, the edge posterior
        $q(z_i = z_j = t) := P(z_i = t, z_j = t | y_i, y_j, x_i, x_j; b^k)$
    by using the MCMC sampling.
    M-step: (Subsection 3.5.2.2)
    Solve the optimization problem
        $\theta^{k+1} := \operatorname*{argmin}_{\theta} Q(\theta; x, y),$
    where $Q(\theta; x, y)$ is defined at Equation 3.12.
    $k \leftarrow k + 1$
until termination criteria are satisfied.
```

and the following edge posterior probability

$$\begin{aligned} & P(z_i = t, z_j = t | y_i, y_j, x_i, x_j; \theta) \\ & = \frac{P(y_i, y_j | x_i, x_j, z_i = t, z_j = t; \theta) P(z_i = t, z_j = t; b)}{\sum\limits_{m,q=1}^{K} P(y_i, y_j | x_i, x_j, z_i = m, z_j = q; \theta) P(z_i = m, z_j = q; b)}, \end{aligned} \tag{3.11}$$

where θ represents the collection of parameters W and b. For small graphs, we can approximate the above distributions using standard MCMC algorithms. Parallel block MCMC can be considered for large graphs.

3.5.2.2 Maximization step

Let us denote with $q(z_i = t) = P(z_i := t | y_i, x_i; \theta)$ and $q(z_i = z_j = t) := P(z_i = t, z_j = t | y_i, y_j, x_i, x_j; \theta)$ the marginal posterior distributions. The M-step of the EM algorithm requires minimizing the negative expected log-likelihood function

$$\begin{aligned} Q(\theta; x, y) := & \sum_{i \in \mathcal{V}} \sum_{t=1}^{K} q(z_i = t) \log \left(1 + e^{-y_i h_{it}(x_i)}\right) \\ & + \lambda \sum_{(i,j) \in \mathcal{E}} \sum_{t=1}^{K} (b_{it} - b_{jt})^2 q(z_i = z_j = t), \end{aligned} \tag{3.12}$$

where θ represents the collection of parameters W and b and $h_{it}(x_i) := W_t^T x_i + b_{it}$. Derivation of this function is given in Subsection 4.2 in the Appendix. For small graphs, standard convex optimization solvers can be used. For large graphs, please refer to Section 3.5.1 where we discuss how we can maximize the expected log-likelihood efficiently with a distributed algorithm for LLGR. The objective function of the E step of LCGR model is the weighted version of the objective function of the LLGR model.

We now comment briefly on the theoretical asymptotic convergence of MCEM to a stationary point of the likelihood function. Convergence theory of MCEM in [86, 30] states that if standard MCMC is used in E step and that the MCMC sample size increases deterministically across MCEM iterations, then MCEM converges almost surely. Parallel block MCMC can be considered for large graphs, then a consequence of blocking of latent variables for the MCMC algorithm is that asymptotic convergence of MCEM is not guaranteed anymore. However, in practice, MCEM is often terminated without even knowing if the algorithm converges to an accurate solution. See for example Section 5 in [86] and references therein about arbitrary termination criteria of MCEM. Therefore, we consider that the parallelism of block MCMC Algorithm offers a trade-off between convergence and computational complexity, which in practice can speed up each iteration of the MCEM algorithm significantly.

3.6 Experiments

In this section, we analyze the empirical performance of the proposed social models on a range of datasets. We outline practical recommendations and illustrate examples where the proposed model is most suitable. The number of iterations of Gibbs sampler in the E-step grows with the number of iterations of the MCEM algorithm. The M-step is implemented using ECOS solver [32] embedded in CVXPY [31] for W updates, bisection line search for b updates, within ADMM iterations.

3.6.1 Illustrative Synthetic Data

We demonstrate that when graph structure and label assignment satisfy certain conditions, our LLGR and LCGR models performs better than other models without social information. We empirically illustrate the impact of graph structure by varying the connectivity between different classes in the graph and illustrate the impact of label assignment by varying the discrepancies of labels in communities of the graph.

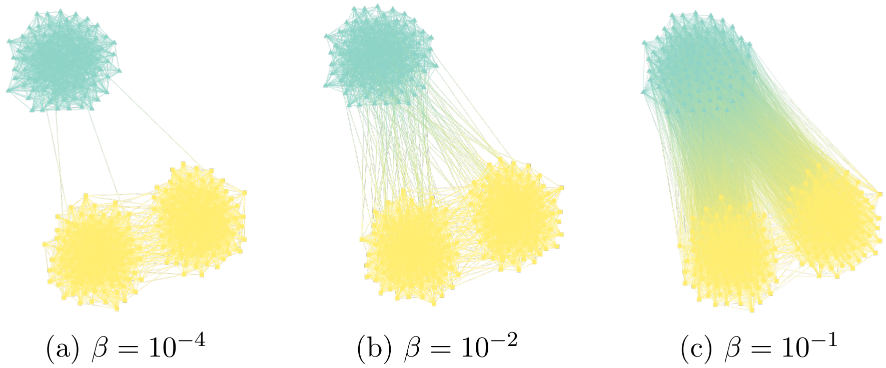

(a) $\beta = 10^{-4}$ (b) $\beta = 10^{-2}$ (c) $\beta = 10^{-1}$

Figure 3.6: The nodes with square shape and yellow color correspond to class $t = 1$. The nodes with triangle shape and turquoise color correspond to class $t = 2$. The larger β the more edges among nodes with different class.

3.6.1.1 Varying connectivity between classes

We consider $N = 300$ individuals and use two different Gaussian distribution to generate the feature vector for each individual. Then we randomly split the individuals into three communities with the same size. We assume that there are two classes, shown in blue and yellow in Figure 3.6. Notice that the feature vectors are assigned to communities regardless of their Gaussian distribution and labels are set based on the classes, which correspond to communities. Therefore, the labels are in align with the graph structure but not in align with the feature space. We set the probability of two individuals that are in the same community to get connected to 0.2, and the probability of two individuals that are in the same class but not in the same community to get connected to 0.01. Then we vary parameter β, the probability of two individuals in different classes to get connected.

Figure 3.6 shows the graph structure when $\beta = 10^{-4}$, $\beta = 10^{-2}$ and $\beta = 10^{-1}$. Notice that the larger β is, the more edges among the communities belong in different classes. Table 3.2 shows the prediction result of four models as a function β. Notice that since feature vectors and labels are not changed as β changes, the prediction of logistic regression and logistic regression with latent class remains constant at 62%. The reason that these models perform poorly is that the labels are not separable given the feature vectors x_i only. Observe in Table 3.2 that when β is as small as 10^{-4}, which means that individuals in different classes are very unlikely to get connected, see Figure 3.6a, the prediction result of the proposed social models is larger than 80%. On the other hand, when β becomes larger, the prediction of the social models is declining. However, as long as $\beta < 0.1$, the proposed LCGR model performs better than logistic regression and logistic regression with latent class models. When $\beta = 0.1$ the social models have the same prediction performance as the logistic regression and latent class models. This is because the classes are not clearly separable on the graph, see Figure 3.6c.

Table 3.2: Prediction results on a randomly chosen test set of 50 individuals when β is varied, i.e., connectivity between classes. For all models the regularization parameter λ which corresponds to the best prediction result out of a range of parameters is chosen.

	β				
model	10^{-4}	10^{-3}	5×10^{-3}	10^{-2}	10^{-1}
logistic reg.	62%	62%	62%	62%	62%
log. reg. lat. class	62%	62%	62%	62%	62%
LLGR	80%	62%	62%	62%	62%
LCGR	88%	82%	64%	62%	62%

3.6.1.2 Varying choice preference parameters

An ideal scenario for the proposed social models is when classes correspond to communities of the given graph and when the labels y_i are clustered according to the classes. However, labels y_i might be misplaced in wrong classes. We study how the preference difference between classes affects the performance of the proposed model.

For this experiment individual feature vectors are generated from a Gaussian distribution with sample size $N = 200$. We randomly split this set of individuals into two parts with the same size, and each part represents a class where individuals share the same parameters W. Assume that W_i is the weight corresponding to the ith group, and $W_1 = -W_2$. For each individual j, b_j is sampled from the same Gaussian distribution. For the graph setting, we set the probability of people in the same class to be connected as 0.2, and the probability of people in different classes to be connected as 10^{-4}. This way, we ensure that classes correspond to communities.

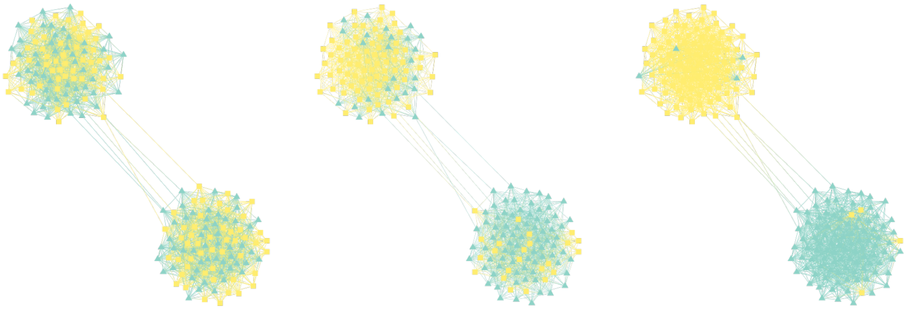
Based on the data generation process, by tuning $\|W_1\|$, we are able to get full control of preference difference among individuals in the two classes. When $\|W_1\|$ becomes larger, preference difference becomes larger as well. As we see in Figure 3.7, when $\|W_1\|$ becomes larger, more individuals in class one have $y_i = 1$ (i.e., yellow squares) and more individuals in class, two have have $y_i = 1$ (i.e., turquoise triangles). When $W_1 = 0$, around half of the individuals in both classes have $y_i = 1$, the other half $y_i = -1$, which means that there is no preference difference between the two classes. Prediction results for this experiment are shown in Table 3.3.

3.6.2 Adolescent smoking

This example uses a dataset collected by [14]. This research program, known as the teenage friends and lifestyle study, has conducted a longitudinal survey of friendships and the emergence of the smoking habit (among other deviant behaviors) in teenage students across multiple schools in Glasgow, Scotland.

Table 3.3: Prediction results on a randomly chosen test set of 50 individuals when $\|W_1\|_2$ is varied. For all models the regularization parameter λ which corresponds to the best prediction result out of a range of parameters is chosen.

	$\|W_1\|_2$					
model	10	5	3	2	1	0
logistic reg.	48%	44%	42%	52%	58%	52%
log. reg. lat. class	48%	48%	54%	54%	42%	36%
LLGR and LCGR	100%	100%	94%	86%	68%	36%

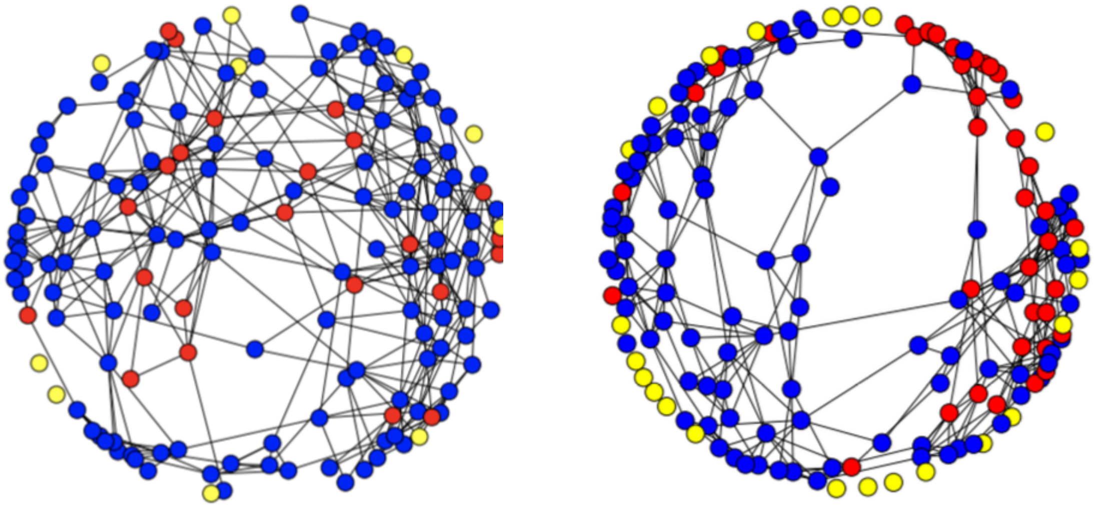


(a) $\|W_1\|_2 = 0$ (b) $\|W_1\|_2 = 4$ (c) $\|W_1\|_2 = 6$

Figure 3.7: Three synthetic examples showing the influence of parameter W_1 on class preference. The nodes with square shape and yellow color correspond to choice $y_i = 1$. The nodes with triangle shape and turquoise color correspond to choice $y_i = -1$.

3.6.2.1 Dataset

Social graphs of 160 students (shown in Figure 3.8) within the same age range of 13-15 years is constructed following a surveyed evidence of reciprocal friendship, with an edge placed among individuals i and j if individual i and individual j named each other friends. We included five variables into the feature vector x_i: age; gender; money: indicating how much pocket money the student had per month, in British pounds; romantic: indicating whether the student is in a romantic relationship; family smoking: indicating whether there were family members smoking at home. Notice that the feature vectors x_i and the edges of the graph are different at different timestamps. The response variable y represents the stated choice that whether a student smokes tobacco ($y_i = 1$), otherwise $y_i = -1$. Note there are nodes with missing labels, but the graph structure should be intact for the parameter estimation of b. Therefore, we set $y_i = 0$ for these nodes, so that the corresponding x_i are not used in the parameter estimation while keeping the graph structure unchanged.



(a) Start of the study (Feb 1995):
127 non-smokers (blue),
23 smokers (red),
10 unobserved (yellow)
422 edges.

(b) End of the study (Jan 1997):
98 non smokers (blue),
39 smokers (red),
23 unobserved (yellow)
339 edges.

Figure 3.8: Social graphs of student friendships and smoking behaviors within the 2 years period of the study.

3.6.2.2 Models comparison

We measured predictions on this dataset and report here 3-fold cross validation results of four models: i) logistic regression; ii) latent class logistic regression; iii) social logistics regression, see Subsection 3.4.3; iv) social latent class logistics regression, see Subsection 3.4.5. Cross-validation process treats the removed nodes as nodes with missing labels, as described above. We consider $K = 2$ latent classes in this experiment. The prediction performance is shown in Tables 3.4.

Table 3.4: Adolescent smoking prediction accuracy, February 1995 and January 1997

model	1995	1997
logistic regression	81.1%	68.5%
latent class	78.9%	72.1%
LLGR	80.0%	76.9%
LCGR	82.2%	80.8%

The high performance of logistic regression and latent class model for the beginning of the study (Table 3.4) indicates that the smoking preferences are largely defined by the individual feature vector. Moreover, parameters in the underlying classes of the latent class model don't differ much. Social latent class model performs equally well. This difference grows significantly when a confounding variable of smoking in the family is removed from the feature list.

Furthermore, by the end of the study (January 1997, Table 3.4 column 2) one can see a significantly higher predictive accuracy of the LLGR and LCGR models. It may indicate that smoking within a certain group of teenagers has become a social norm (indicating the difference in offset parameters b_{it}), or that the response y_i to independent factors x_i within that group differs from the others as reflected by differences in W_t.

Notice the significant decrease in prediction accuracy for nonsocial models between column 1 and column 2 in Table 3.4. This is because at the beginning of the study (February 1995) less than 15% of teenagers smoked, while at the end of the study (January 1997) about 25% of teenagers smoked. This difference is likely due to social norms developed among teenagers that are captured by the graph and therefore missed by nonsocial models.

3.6.2.3 Parameter visualization

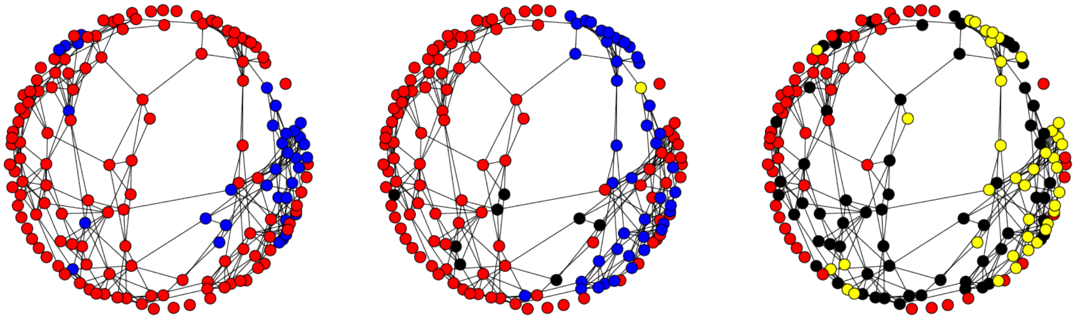
We are going to illustrate how the specification of the proposed model allows in-depth exploration of the parameters to assist in making this type of conclusions. To that end, we are going to explore parameters b_{it} and class membership probabilities across the regularization path of hyper-parameter λ.

The estimated class membership (the probability of being in a given latent class) on the graph is shown in Figure 3.9. A visualization of estimated b_{it} of one latent class for several values of λ is shown in Figure 3.10. When $\lambda = 0.1$, smoking pattern begins to show across the graph, i.e., compare Figures 3.8 and 3.10b. Let us note that $\lambda = 0.1$ value corresponds to the best prediction accuracy in our experiments for the proposed social latent class logistic regression model for the smoke data at the end of the study (January 1997). This is because the social latent class logistic regression model is able to clearly distinguish a group of socially connected individuals within which the choice preferences towards smoking are higher.

When $\lambda = 0.01$, b_{it} are similar across the nodes. On the other hand, when $\lambda = 10$, b_{it} are not similar across the nodes. Although this is counter-intuitive since the graph regularization favors similar b_{it} across nodes for large values of λ, it is explained by the node and edge posterior distribution in the M-step which also controls regularization across nodes during the execution of the algorithm.

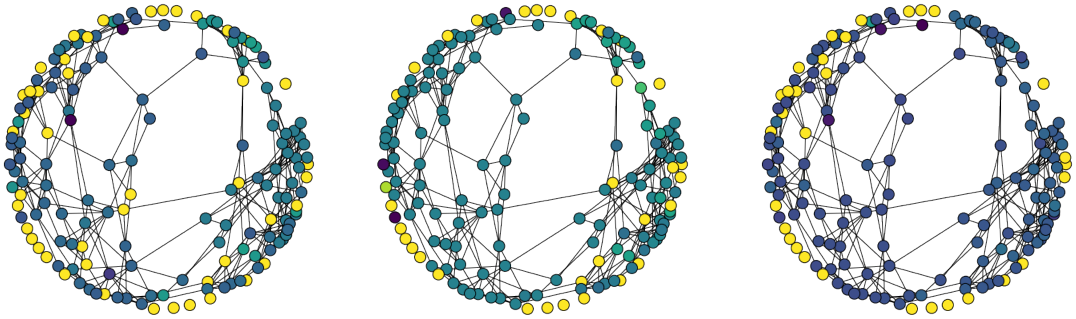
3.6.3 Online Retail Account Relation Graph Data

In this section, we experimented our model on a real world online retail account relation graph from a leading commercial company in the US, and showed that our model excelled in fraud account prediction task when compared with other models. **More details on the data and the empirical results have been redacted until their release has been approved**



(a) $\lambda = 10$ (b) $\lambda = 1$ (c) $\lambda = 0.01$

Figure 3.9: Class membership probabilities estimated for the nodes for multiple values of λ. Blue, yellow, green, black and red colours correspond to probabilities about 0, $(0, 0.5)$, 0.5, $(0.5, 1)$ and 1, respectively.



(a) $\lambda = 10$ (b) $\lambda = 1$ (c) $\lambda = 0.01$

Figure 3.10: The values of b_{it} estimated for the nodes for multiple values of λ. Lighter color corresponds to higher values.

3.7 Conclusions and Future Work

In this paper, we introduced social graph regularization ideas into discrete choice models for user choice modeling. We proposed local logistics graph regularization (LLGR) method and latent class graph regularization (LCGR) model. We developed scalable parameter estimation method for LLGR model on large graphs benefiting from recent advances in distributed optimization based on ADMM methods. Also, we have developed, implemented, and explored parameter estimation algorithms that allow parallel processing implementation for both E- and M-steps of the Monte Carlo Expectation Maximization (MCEM) algorithms for LCGR model. In experimental evaluation, we have focused on investigating the usefulness of the models in revealing and supporting the hypothesis in studies where not only predictive performance (that was found to be highly competitive), but also understanding social influence, is crucial. Our models can be directly applied to study social influence on revealed choices in large social graphs with rich node attributes. One challenge with extending our results is that such data are very rarely available in open access due to privacy issues.

Chapter 4

Social Discrete Choice Models Appendix

4.1 Improving E step

Here, we describe how to scale up the E step in Section 3.5.2.1, which is important for MCEM method, as described in Section 3.5.2. In the LCGR model, we use a markov random field to model the joint distribution of latent class conditioned on local coefficients b. And we need to calculate the posterior node probabilities (Equation (3.10)) and edge probabilities (Equation (3.11)) in the E step based on Equation (3.7). Let us consider the case of two classes as an example. Assume the labels of the two classes are 1 and -1, then we can simplify Equation (3.7) to

$$P(z;b) \propto \prod_{(i,j)\in\mathcal{E}} \exp((\frac{\theta_{ij}-\beta_{ij})z_i}{4} + \frac{(\theta_{ij}+\beta_{ij})z_j}{4} + \frac{(\theta_{ij}+\beta_{ij})z_i \times z_j}{4} + \frac{(\theta_{ij}+\beta_{ij})}{4})$$

where we have:

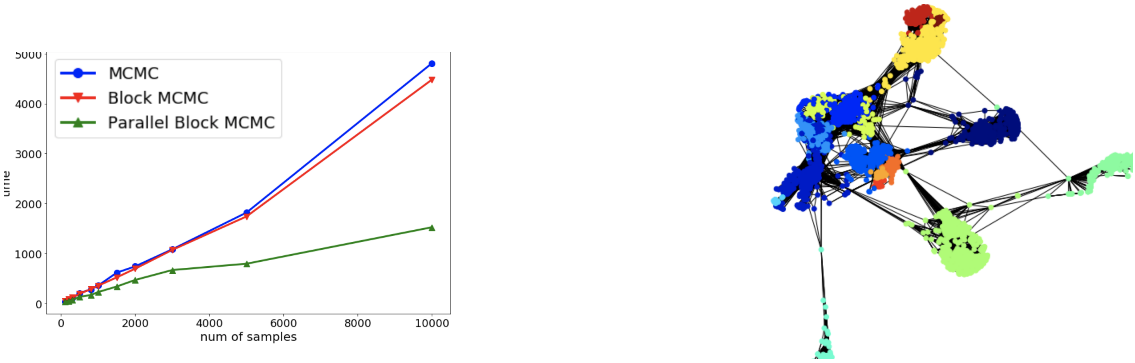
$$\theta ij = (b_{ik} - b_{jk})^2, k = 1$$
$$\beta ij = (b_{ik} - b_{jk})^2, k = -1$$

And the edge potentials are defined as below:

$$\exp((\frac{\theta_{ij}-\beta_{ij})z_i}{4} + \frac{(\theta_{ij}+\beta_{ij})z_j}{4} + \frac{(\theta_{ij}+\beta_{ij})z_i \times z_j}{4} + \frac{(\theta_{ij}+\beta_{ij})}{4})$$

Thus, our model belongs to the standard pairwise Markov Random Field.

Both variational methods and sampling-based methods are suitable for our problem setting. (1) For variational methods, the exact inference of Markov Random Field using Bethe approximation [126, 119] can calculate distributions of nodes and edges. The approximate inference algorithms, e.g., mean field inference [119] and loopy belief propagation (BP) [85], although exhibiting excellent empirical performance, $P(z_i = t, z_j = t; b)$ is not calculated. (2) For sampling-based approach, MCMC algorithms can calculate both the node and edge distributions.



(a) Parallel block MCMC computation time

(b) Facebook Ego Network Community Detection Result

Figure 4.1: Left: Visualization of Facebook Ego Graph, where color represents community membership. Right: plot of number of samples VS running time, for standard MCMC, block MCMC and parallel block MCMC

Calculating the marginal posterior probabilities Equation (3.10) and Equation (3.11) is computationally expensive due to the marginal probabilities $P(z_i = t; b)$ and $P(z_i = t, z_j = t; b)$. This is because to calculate the latter two, we have to marginalize $N-1$ and $N-2$ latent variables, respectively. We chose sampling-based method and accelerated the computation by using a block MCMC sampling technique to compute $P(z_i = t; b)$ and $P(z_i = t, z_j = t; b)$. The algorithm uses a preprocessing step to partition the graph into c disjoint communities. Then it runs an MCMC algorithm on each community/block in *parallel* by ignoring the edges among the blocks.

Figure 4.1a demonstrates how parallel block MCMC speed up the computation. We use the Facebook Ego Network Data as shown in Figure 4.1b, where there are 4039 nodes and 88234 edges. We randomly generate the local b value of each class for all the nodes. We first run Louvain community detection algorithm [10], the state-of-the-art greedy optimization algorithm for global community detection, and extract 10 communities from the graph. We then run MCMC/ Block MCMC and Parallel Block MCMC and time the code for each iteration. As can be seen from Figure 4.1a, the performance of MCMC and Block MCMC are almost the same, because the only difference is that Block MCMC omits the edges between communities for the Gibbs update. Since the MCMC update of each community is independent of each other, we can adopt the parallel computing paradigm for Block MCMC. As we can see from Figure 4.1a, Parallel Block MCMC performs much better than the other two.

4.2 Negative expected log-likelihood in Eq. (8)

We denote with $q(z) := P(z|y, x; \theta)$ the posterior distribution, with $\sum_z$ the sum over all latent variables z, and θ represents the collection of parameters W and b. Let $\mathbf{1}(z_i = t)$ be the indicator function, which is equal to 1 if $z_i = t$. We assume that z follows

$$P(z;b) \propto \prod_{(i,j)\in\mathcal{E}} \exp\left(-\lambda \sum_{t=1}^{K} (b_{it} - b_{jt})^2 \mathbf{1}(z_i = z_j = t)\right), \tag{4.1}$$

and

$$P(y_i|x_i, z_i = t, \theta) = 1/(1 + e^{-y_i h_i(x_i)}), \tag{4.2}$$

where $h_{it}(x_i) := W_t^T x_i + b_{it}$. The derivation of the expected log-likelihood is shown below.

$$\begin{aligned}
\tilde{Q}(\theta; x, y) &:= \sum_z q(z) \log P(y, z|x; \theta) \\
&= \sum_z q(z) \log(P(y|x, z; \theta) P(z; b)) \\
&= \sum_z q(z) \log P(y|x, z; \theta) + \sum_z q(z) \log P(z; b) \\
&= \sum_z q(z) \log \left(\prod_{i\in\mathcal{V}} \sum_{t=1}^{K} P(y_i|x_i, z_i = t, \theta) \mathbf{1}(z_i = t)\right) \\
&= \sum_z q(z) \sum_{i\in\mathcal{V}} \log \left(\sum_{t=1}^{K} P(y_i|x_i, z_i = t, \theta) \mathbf{1}(z_i = t)\right) \\
&\quad - \lambda \sum_z q(z) \sum_{(i,j)\in\mathcal{E}} \sum_{t=1}^{K} (b_{it} - b_{jt})^2 \mathbf{1}(z_i = z_j = t).
\end{aligned}$$

We can exchange the sequence of log and $\sum_{t=1}^{K}$ because each node can only be in one class, thus we have

$$\begin{aligned}
\tilde{Q}(\theta; x, y) &= \sum_z q(z) \sum_{i\in\mathcal{V}} \sum_{t=1}^{K} \mathbf{1}(z_i = t) \log P(y_i|x_i, z_i = t; \theta) \\
&- \sum_z q(z) \sum_{(i,j)\in\mathcal{E}} \sum_{t=1}^{K} \lambda (b_{it} - b_{jt})^2 \mathbf{1}(z_i = z_j = t).
\end{aligned}$$

Let's write the summation over z inside the summation over vertices and the summation over latent variables

$$\begin{aligned}
\tilde{Q}(\theta; x, y) &= \sum_{i\in\mathcal{V}} \sum_{t=1}^{K} \sum_z q(z) \mathbf{1}(z_i = t) \log P(y_i|x_i, z_i = t; \theta) \\
&- \lambda \sum_{(i,j)\in\mathcal{E}} \sum_{t=1}^{K} \sum_z q(z) (b_{it} - b_{jt})^2 \mathbf{1}(z_i = z_j = t),
\end{aligned}$$

and then we get the marginal probabilities

$$
\begin{aligned}
\tilde{Q}(\theta; x, y) = & \sum_{i \in \mathcal{V}} \sum_{t=1}^{K} q(z_i = t) \log P(y_i | x_i, z_i = t, \theta) \\
& - \lambda \sum_{(i,j) \in \mathcal{E}} \sum_{t=1}^{K} (b_{it} - b_{jt})^2 q(z_i = z_j = t).
\end{aligned}
$$

Using (4.2) and multiplying by minus equation $\tilde{Q}(\theta; x, y)$ we get the negative expected log-likelihood function in (3.12).

4.3 Discrete Choice Models Review

We have used logistics regression and latent class models as benchmarks for the Local Logistics Graph Regularization (LLGR) and Latent Class Graph Regularization (LCGR) models. In this section, we focus on the specifications and estimation of the two models.

4.3.1 Random Utility Models and Multinomial Discrete Choice Models

Random utility models in discrete choice analysis is based on the assumption that the utility, associated with each available alternatives in the consideration set, is stochastic. As shown in structural equation 4.3, random utility of alternative n for individual i ($U_{i,n}$), consists of the two parts: the observed deterministic component $V_{i,n}$ and the errors of the utilities $\epsilon_{i,n}$ [121, 84]. It is assumed that the observed deterministic component $V_{i,n}$ is a function of feature vector $X_{i,n}$ with parameter β.

$$
U_{i,n} = V_{i,n} + \epsilon_{i,n} \tag{4.3}
$$

The random utility models also conform to the utility maximization decision rule: the decision maker will choose the alternative that maximizes his/her utility. C_i denotes the choices set available to decision maker individual i.

$$
Y_i = n, \text{if } U_{i,n} = \max_{m \in C_i} U(i, m) \tag{4.4}
$$

McFadden [80] derived the multinomial logit model (MNL) from the random utility framework by making below assumptions:

- All $\epsilon_{i,n}$ are independent and identically distributed (IID) following Gumbel extreme value distribution.
- The location parameter η of the Gumbel extreme value distribution is set to 0, and the scale parameter μ is set to 1.

These assumptions lead to the probability distribution of individual choices 4.5.

$$P(Y_i = n|X_i;\theta) = \frac{e^{V(X_{i,n};\theta)}}{\sum_{m \in C_i} e^{V(X_{i,m};\theta)}} \tag{4.5}$$

The binary logit (BNL) model, without alternative specific attributes, is equivalent to the logistics regression model. While the multinomial logit (MNL) model, without alternative specific attributes, is equivalent to the softmax regression model. This is how the discrete choice analysis theory in the social science, is connected to machine learning in computer science.

4.3.2 Latent Class Models

4.3.2.1 Assumption and log-likelihood objective function

Latent class model was introduced into the discrete choice analysis to overcome the limitations of multinomial logit model (MNL), most notably the independence from irrelevant alternatives property (IIA). The underlying assumption for the latent class model is that choices are dependent on observed attributes (feature vectors) and unobserved latent heterogeneity among individuals. For example, taste differences and decision protocols. This heterogeneity is analyzed by approximating a continuous distribution with a discrete distribution through latent class discrete variables. It is assumed that the probability distribution for individual i, conditioned on the individual belongs to a latent class k, are independent. Assume that conditional probability distribution is $P(Y_i|x_i, Z_i = k, \theta)$, then the probability distribution of the choice of individual i is as below 4.6. Assume that we don't have any assumption on the distribution of the latent class variables for now :

$$P(Y_i = n|x_i, \theta) = \sum_{k=1}^{K} \mathbf{1}(z_i = k) P(Y_i = n|x_i, z_i = k, \theta) \tag{4.6}$$

Based on this probability distributions, the objective function of the maximum likelihood estimation (MLE) can be rewritten as below. However, because there is summation within the logarithmic operation, the direct optimization could be difficult and highly unstable. Instead of directly optimizing the parameters, we convert the MLE problem to another easier optimization problem. Jensen's Inequality is used to derive the lower bound of the log likelihood. The lower bound which is also called the auxiliary, contains $q(z)$ and the complete log likelihood $\log P(y, x, z; \theta)$. Note here $q(z)$ is an arbitrary distribution q over the latent variables space $\mathcal{Z}$. The MLE parameter estimation problem of latent class model is transformed into finding optimal $q(z)$ and θ.

$$\begin{aligned}\log P(y;x,\theta) &= \log \sum_z P(y,x,z;\theta) = \log \sum_z q(z)\frac{P(y,x,z;\theta)}{q(z)}\\ &= \log E_{z\sim q}\left[\frac{P(y,x,z;\theta)}{q(z)}\right]\\ &\geq \sum_z q(z)\log\frac{P(y,x,z;\theta)}{q(z)} \equiv \mathcal{L}(q,\theta)\\ &= \sum_z -q(z)\log q(z) + \sum_z q(z)\log P(y,x,z;\theta).\end{aligned}$$

The Expectation-Maximization (EM) algorithm proceeds by coordinate ascent algorithm in which we alternatively update the $q(z)$ distribution and θ, to find the optimal $q(z)$ and θ. In the E step, the target is to find the optimal $q(z)$ given the estimated θ. In the M step, the target is to update the value of parameters given the updated $q(z)$ distribution. The EM algorithm alternates the E step and the M step until convergence.

4.3.2.2 EM algorithm

In the Expectation step (E-step), we form the following optimization problem to to find the optimal distribution $q(z)$

$$\max_{q(z)} \sum_z -q(z)\log q(z) + \sum_z q(z)\log P(y,x,z;\theta).$$

Let's take the gradient of the objective function and set it to zero

$$\begin{aligned}\nabla_{q(z)} &= \sum_z -\log q(z) - 1 + \log P(y,x,z;\theta)\\ &= \sum_z \left(\log\frac{P(y,x,z;\theta)}{q(z)} - 1\right)\\ &= 0.\end{aligned}$$

To solve this optimization problem, $q(z)$ must be such that $\log(\frac{P(y,x,z;\theta)}{q(z)})$ is constant, and this is satisfied by setting, which means the optimal distribution $q(z)$ is the posterior distribution of latent variables.

$$q(z) = \frac{P(y,x,z;\theta)}{\sum_{\dot{z}} P(y,x,\dot{z};\theta)} = P(z|y,x;\theta).$$

Let's denote τ_{it} as the posterior distribution defined as below:

$$\tau_{it} = P(z_i = t, y_i|\theta, x_i)/\sum_{j=1}^{k} P(y_i|z_i = j;\theta, x_i)P(z_i = j)$$

In the Maximization step (M-step) we form the following optimization problem

$$\max_{\theta} \sum_{z} -q(z) \log q(z) + \sum_{z} q(z) \log P(y, x, z; \theta).$$

Since the first part of the summation is not related to θ our goal is to maximize the expectation of log likelihood with respect to θ, therefore we will focus on the second part.

$$E_{z \sim q}[\log P(y, x, z; \theta)] \equiv \sum_{z} q(z) \log P(y, x, z; W, b)$$

Notice here we have another important assumption on the distribution of latent class variables: And we assume the latent class variables are independent and identically distributed (IID) following the multinomial distribution. That is,

$$z_i \sim \text{Mult}(\pi, 1), \forall i$$

Given the assumption on the latent class variables probabilities distribution, we can write $P(y_i, z_i; \theta)$ as

$$P(y_i, z_i; \theta) = \sum_{t=1}^{k} P(y_i | z_i = t, \theta) \pi_t$$

Assume the parameters are $\theta = (w, b, \pi)$. Then the optimization problem of the M step is formulated as below:

$$\max_{w,b,\pi} \sum_{i \in V} \sum_{t=1}^{K} \log(y_i | z_i = t, \theta) \tau_{it} + \sum_{i \in V} \sum_{t=1}^{K} \tau_{it} \log \pi_t$$
$$s.t. \sum_{t=1}^{K} \pi_t = 1$$

Note that the objective function is separable regarding t, so $\forall t$, a sub optimization problem is formulated as below:

$$\max_{w,b} \sum_{t=1}^{K} \log(y_i | z_i = t, \theta) \tau_{it} + \sum_{i \in V} \tau_{it} \log \pi_t$$

Since this objective function is concave in w_t and b_t and differentiable, we can use CVXPY [31] convex optimization solver to find optimal W and b. To update π, we need to solve the following linearly constrained optimization problem

$$\max_{\pi} \sum_{i \in V} \log(\pi_t) \tau_{it}, \forall t \in [1, K]$$
$$s.t. \sum_{t=1}^{K} \pi_t = 1$$

The optimal solution for this problem is $\pi_t = \frac{1}{N}\sum_{i=1}^{N}\tau_{it}$.

The code based on the derivation is implemented in Python using numpy library and CVXPY [31] library. The implementation is available at https://github.com/DanqingZ/social-DCM.

4.4 Alternating Direction Method of Multiplier (ADMM) Algorithm for Local Logistics Graph Regularization (LLGR) model

The objective function of the Local Logistics Graph Regularization (LLGR) model 3.8 consists of local variables on each node.

$$Q(\theta; x, y) := \sum_{i\in\mathcal{V}} \log\left(1 + e^{-y_i h_i(x_i)}\right) + \lambda \sum_{(i,j)\in\mathcal{E}} (b_i - b_j)^2 \tag{4.7}$$

Although the objective function is convex, stochastic gradient descent cannot be simply applied for the parameter estimation. In section 3.5.1, we proposed to use an easy-to-implement algorithm based on the Alternating Direction Method of Multiplier. In this section, we focus on the motivation, derivation and formulation for the Alternating Direction Method of Multiplier (ADMM) Algorithm for Local Logistics Graph Regularization (LLGR) model.

4.4.1 Introduction of ADMM

ADMM is a simple yet powerful algorithm that solves the large-scale distributed optimization problem in a distributed and scalable manner. It is an algorithm that blends the benefits of two earlier approaches, which are dual decomposition algorithm [37, 68] and augmented Lagrangian algorithms for the constrained optimization problem [12].

Dual decomposition method is based on the dual ascent method. In the dual ascent method, we run gradient descent on the dual problem instead of running projected gradient descent on the primal problem. The dual ascent method consists of two steps, the primal variables optimization steps, and the dual variables update step. The benefit of the dual ascent method is that when the primal objective function is separable, the dual ascent method can lead to a scalable decentralized algorithm, which means the primal variables optimization steps can be implemented in parallel. This kind of dual ascent method is called the **dual decomposition method**.

The other related method is the **augmented Lagrangian method**, which was developed to add robustness to the dual ascent method. The first step of the augmented Lagrangian method is to define the augmented Lagrangian, where we add an extra term to the objective function as the penalty term. The second step is to run the dual ascent algorithm on the augmented Lagrangian, which is called the method of multipliers algorithm. It is straightforward to blend the method of multipliers with the dual decomposition when the objective function is separable. And this how the alternating direction method of multiplier method is motivated.

The canonical form of problem that ADMM solves is as below, where f and g are convex functions, x and z are decision variables:

$$\begin{aligned} \min_{x} \quad & f(x) + g(z) \\ \text{subject to} \quad & Ax + Bz = c \end{aligned}$$

Then the augmented Lagrangian can be formed as below, where penalty term is added to the Lagrangian for the primal problem.

$$L_\rho(x, y, r) = f(x) + g(z) + r^T(Ax + Bz - c) + (\rho/2)\left\|Ax + Bz - c\right\|_2^2$$

Then we can run gradient ascent algorithm. And since the objective function of the primal problem is decomposable into two parts, the primal variables optimization steps can be implemented in parallel. ADMM algorithm consists of below steps in each iteration.

$$\begin{aligned} x^{k+1} &= \arg\min_x L_\rho(x, z^k, r^k) \\ z^{k+1} &= \arg\min_z L_\rho(x^{k+1}, z, r^k) \\ r^{k+1} &= r^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \end{aligned}$$

ADMM algorithms has the advantages of both dual decomposition and augmented Lagrangian, that it is a robust and scalable decentralized algorithm.

We refer the reader to [12, 44] for a detailed discussion of dual ascent, dual decomposition, augmented Lagrangian, ADMM scaled form and ADMM convergence rates.

4.4.2 ADMM for the LLGR model

Following the work of [50] that applies ADMM to network lasso method, we aim to use the ADMM algorithm in a way that each node solves it own optimization problem and then pass the output of its optimization problem to its neighbors, update the values of the dual variables, and repeats until convergence. The difference between our model and the network lasso problem [50] is that the network lasso problem assumes the optimization problem only has local variables. However, our model assumes that there are both local variables and global variables in the optimization problem, which is a more general case. The first step of our algorithm is to convert the unconstrained optimization problem into a equality-constrained optimization problem. In this way, the objective function of the primal optimization problem is separable for each node and each edge.

$$\begin{aligned} \min_{W,b} \quad & \sum_{i\in\mathcal{V}} \log\left(1 + e^{-y_i(g_i^T x_i + b_i)}\right) + \lambda \sum_{(i,j)\in\mathcal{E}} (s_{ij} - s_{ji})^2 \\ \text{s.t:.} \quad & b_i = s_{ij} \;\; j \in \mathcal{N}(i), \;\; \forall i \\ & W = g_i, \;\; \forall i, \end{aligned} \tag{4.8}$$

Based on the equality-constrained optimization problem, we can define the augmented Lagrangian below, where u and r are the dual variables and ρ_1 and ρ_2 are the penalty parameters. Note this augmented Lagrangian is written slightly different from 4.4.1, which is called the unscaled form of augmented Lagrangian [12]. In our ADMM algorithm, we use the scaled form of augmented Lagrangian [12], because it is shorter and easier to demonstrate. We define the augmented Lagrangian below, where u and r are the dual variables and ρ_1 and ρ_2 are the penalty parameters, λ is the regularization hyperparameter.

$$
\begin{aligned}
L_{\rho,\lambda}(W, b, g, s, u, r) &:= \sum_{i \in \mathcal{V}} \Big\{ \log \Big(1 + e^{-y_i(g_i^T x_i + b_i)} \Big) \\
&+ \frac{\rho_1}{2} \Big(- \|r_i\|_2^2 + \|W - g_i + r_i\|_2^2 \Big) \Big\} + \sum_{(i,j) \in \mathcal{E}} \Big\{ \lambda (s_{ij} - s_{ji})^2 \\
&+ \frac{\rho_2}{2} \Big(- \|u_{ij}\|_2^2 - \|u_{ji}\|_2^2 + \|b_i - s_{ij} + u_{ij}\|_2^2 + \|b_j - s_{ji} + u_{ji}\|_2^2 \Big) \Big\}
\end{aligned}
$$

Based on the scaled-form augmented Lagrangian, we can derive the ADMM algorithm as below:

Algorithm 3 ADMM for Problem 3.9

```
Initialize:
    $k \leftarrow 0$, $W^k$, $b^k$, $g^k$, $s^k$, $u^k$ and $r^k$
repeat
    Set $W_t^{k+1} = \frac{1}{N}\sum_{i=1}^{N}(g_i^k - r_i^k)$
    $b_{it}^{k+1} := \arg\min_{b_i} \; L_\rho(W_t^{k+1}, b_i, g^k, s^k, u^k, r^k) \; \forall i \in \mathcal{V}$
    $g_i^{k+1} := \arg\min_{g_i} \; L_\rho(W^{k+1}, b^{k+1}, g_i, s^k, u^k, r^k) \; \forall i \in \mathcal{V}$
    $s_{ij}^{k+1} = \frac{(2\lambda+\rho_2)(b_i+u_{ij})+2\lambda(b_j+u_{ji})}{4\lambda+\rho_2}$
    $s_{ji}^{k+1} = \frac{(2\lambda+\rho_2)(b_j+u_{ji})+2\lambda(b_i+u_{ij})}{4\lambda+\rho_2} \; \forall (i,j) \in \mathcal{E}$
    Set
        $r_i^{k+1} = r_i^k + \rho_1(W^{k+1} - g_i^{k+1}) \; \forall i \in \mathcal{V}$
        $u_{ij}^{k+1} = u_{ij}^k + \rho_2(b_i^{k+1} - s_{ij}^{k+1}) \; \forall (i,j) \in \mathcal{E}$
        $u_{ji}^{k+1} = u_{ji}^k + \rho_2(b_j^{k+1} - s_{ji}^{k+1})$
    $k \leftarrow k + 1$
until termination criteria are satisfied.
```

Note updating s_{ij} and s_{ji} procedure can be formulated as an optimization below:

$$
s_{ij}^{k+1}, s_{ji}^{k+1} = \arg\min_{s_{ij}, s_{ji}} \; f(s_{ij}, s_{ji}) \; \forall (i,j) \in \mathcal{E} \tag{00}
$$

It has closed form solution:

$$\begin{aligned} s_{ij}^{k+1} &= \frac{(2\lambda + \rho_2)(b_i + u_{ij}) + 2\lambda(b_j + u_{ji})}{4\lambda + \rho_2} \\ s_{ji}^{k+1} &= \frac{(2\lambda + \rho_2)(b_j + u_{ji}) + 2\lambda(b_i + u_{ij})}{4\lambda + \rho_2} \end{aligned}$$

For the weighted version of the LLGR model as below:

$$Q(\theta; x, y) := \sum_{i \in \mathcal{V}} \log\left(1 + e^{-y_i h_i(x_i)}\right) + \lambda \sum_{(i,j) \in \mathcal{E}} q_{ij}(b_i - b_j)^2 \tag{4.9}$$

where q_ij is the weight on edge e_{ij} representing the closeness between individual i and individual j. For friends with similar feature vector, the closer they are, the more similar their probability distributions are.

Based on the objective function, we can conduct similar procedure to convert the unconstrained optimization problem and write out the augmented Lagrangian for the constrained optimization problem.

$$\begin{aligned} L_{\rho,\lambda}(W, b, g, s, u, r) &:= \sum_{i \in \mathcal{V}} \Big\{ \log\Big(1 + e^{-y_i(g_i^T x_i + b_i)}\Big) \\ &+ \frac{\rho_1}{2}\Big(- \|r_i\|_2^2 + \|W - g_i + r_i\|_2^2\Big)\Big\} + \sum_{(i,j) \in \mathcal{E}} \Big\{\lambda q_{ij}(s_{ij} - s_{ji})^2 \\ &+ \frac{\rho_2}{2}\Big(- \|u_{ij}\|_2^2 - \|u_{ji}\|_2^2 + \|b_i - s_{ij} + u_{ij}\|_2^2 + \|b_j - s_{ji} + u_{ji}\|_2^2\Big)\Big\} \end{aligned}$$

The ADMM algorithm for the weighted LLGR model is similar to the unweighted version. The difference is in the update rule of s_{ij} and s_{ji}. The closed form solution of the s_{ij} and s_{ji} update are as follows:

$$\begin{aligned} s_{ij}^{k+1} &= \frac{(2\lambda q_{ij} + \rho_2)(b_i + u_{ij}) + 2\lambda q_{ij}(b_j + u_{ji})}{4\lambda q_{ij} + \rho_2} \\ s_{ji}^{k+1} &= \frac{(2\lambda q_{ij} + \rho_2)(b_j + u_{ji}) + 2\lambda q_{ij}(b_i + u_{ij})}{4\lambda q_{ij} + \rho_2} \end{aligned}$$

As we can see from above, the ADMM algorithm mainly consists of two blocks, the primal variables optimization, and the dual variables update via dual ascent. “Alternating" means that we can alternately update the values of the primal and dual variables in each iteration until convergence. In each iteration, the computation of each node is fully distributed so that the LLGR method can be fully parallel.

4.5 Detailed Proof of Expectations Maximization Algorithm for the Latent Class Graph Regularization (LCGR) Model

4.5.1 Lower bound

Based on the assumption of the conditional probabilities distribution given latent class 3.3, and the joint probability distribution of the latent class variables 3.7. The objective function of the maximum likelihood estimation (MLE) can be rewritten as below. Similar to the first step of the latent class model, we convert the optimization problem into another optimization problem eliminating discrete variables z_i. Let z denote the vector of hidden variables for all individuals which follows some distribution $z \sim q(z)$. Moreover, we denote with θ all the parameters of our model. We can use Jensen's inequality to derive lower bound for the log likelihood of our model

$$\begin{aligned}\log P(y;x,\theta) &= \log \sum_z P(y,x,z;\theta) = \log \sum_z q(z)\frac{P(y,x,z;\theta)}{q(z)} \\ &= \log E_{z\sim q}\left[\frac{P(y,x,z;\theta)}{q(z)}\right] \\ &\geq \sum_z q(z)\log\frac{P(y,x,z;\theta)}{q(z)} \\ &= \sum_z -q(z)\log q(z) + \sum_z q(z)\log P(y,x,z;\theta).\end{aligned}$$

4.5.2 Expectation step

In the Expectation step (E-step), we form the following optimization problem to to find the optimal distribution $q(z)$

$$\max_{q(z)} \sum_z -q(z)\log q(z) + \sum_z q(z)\log P(y,x,z;\theta).$$

Let's take the gradient of the objective function and set it to zero.

$$\begin{aligned}\nabla_{q(z)} &= \sum_z -\log q(z) - 1 + \log P(y,x,z;\theta) \\ &= \sum_z \left(\log\frac{P(y,x,z;\theta)}{q(z)} - 1\right) \\ &= 0.\end{aligned}$$

To solve this optimization problem, $q(z)$ must be such that $\log(\frac{P(y,z;x,\theta)}{q(z)})$ is constant, and this is satisfied by setting

$$q(z) = \frac{P(y,z;x,\theta)}{\sum_{\dot{z}} P(y,\dot{z};x,\theta)} = P(z|y;x,\theta).$$

This means that the optimal distribution $q(z)$ is the posterior distribution of latent variables.

4.5.3 Maximization step and the expected complete log likelihood

In the Maximization step (M-step) we form the following optimization problem

$$\max_{\theta} \sum_{z} -q(z)\log q(z) + \sum_{z} q(z) \log P(y,z;x,\theta).$$

Since the first part of the summation is not related to θ our goal is to maximize the expectation of log likelihood with respect to θ, therefore we will focus on

$$\begin{aligned}
E_{z\sim q}[\log P(y,z;x,\theta)] &\equiv \sum_{z} q(z)\log P(y,z;x,W,b) \\
&= \sum_{z} q(z)\log(P(y|z;x,W,b)P(z;W,b)) \\
&= \sum_{z} q(z)\log P(y|x,z;W,b) + \sum_{z} q(z)\log P(z;W,b) \\
&= \sum_{z} q(z)\log\left(\prod_{i\in V}\sum_{k=1}^{K} P(y_i|x_i,z_i=k,\theta)\mathbf{1}(z_i=k)\right) \\
&- \lambda \sum_{z} q(z) \sum_{(i,j)\in\mathcal{E}} \sum_{k=1}^{K} \left((b_{ik}-b_{jk})^2 + \|w_{ik}-w_{jk}\|_2^2\right)\mathbf{1}(z_i=z_j=k) \\
&= \sum_{z} q(z)\sum_{i\in V}\log\left(\sum_{k=1}^{K} P(y_i|x_i,z_i=k,\theta)\mathbf{1}(z_i=k)\right) \\
&- \lambda \sum_{z} q(z) \sum_{(i,j)\in\mathcal{E}} \sum_{k=1}^{K} \left((b_{ik}-b_{jk})^2 + \|w_{ik}-w_{jk}\|_2^2\right)\mathbf{1}(z_i=z_j=k).
\end{aligned}$$

We can exchange the sequence of log and $\sum_{t=1}^{K}$ because each node can only be in one class, thus we have

$$\begin{aligned}
E_{z\sim q}[\log P(y,x,z;\theta)] &= \sum q(z)\sum_{i\in V}\sum_{k=1}^{K}\mathbf{1}(z_i=k)\log P(y_i|x_i,z_i=k,\theta) \\
&- \lambda \sum_{z} q(z) \sum_{(i,j)\in\mathcal{E}} \sum_{k=1}^{k} \left((b_{ik}-b_{jk})^2 + \|w_{ik}-w_{jk}\|_2^2\right)\mathbf{1}(z_i=z_j=k).
\end{aligned}$$

Let's write the summation over z inside the summation over vertices and summation over latent classes

$$\begin{aligned} & E_{z\sim q}[\log P(y,x,z;\theta)] \\ &= \sum_{i\in V}\sum_{k=1}^{K}\sum_{z} q(z_1,\dots,z_i,\dots,z_n)\mathbf{1}(z_i=k)\log P(y_i|x_i,z_i=k,\theta) \\ &-\lambda\sum_{(i,j)\in\mathcal{E}}\sum_{k=1}^{K}\sum_{z} q(z_1,\dots,z_i,\dots,,z_j,\dots,z_n)\left((b_{ik}-b_{jk})^2+\|w_{ik}-w_{jk}\|_2^2\right)\mathbf{1}(z_i=z_j=k). \end{aligned}$$

Notice that we can pull $\log(P(y_i|x_i,z_i=k,\theta))$ outside because the probability takes as input the class and the index i. On the other hand the indicator function $\mathbf{1}(z_i=k)$ cannot be pulled outside because the indicator function is a function of z_i. Therefore we have that

$$\begin{aligned} E_{z\sim q}[\log P(y,x,z;\theta)] &= \sum_{i\in V}\sum_{k=1}^{K}\log(P(y_i|x_i,z_i=k,\theta))\sum_{z} q(z)\mathbf{1}(z_i=k) \\ &-\lambda\sum_{(i,j)\in\mathcal{E}}\sum_{k=1}^{K}\left((b_{ik}-b_{jk})^2+\|w_{ik}-w_{jk}\|_2^2\right)\sum_{z} q(z)\mathbf{1}(z_i=z_j=k) \\ &= \sum_{i\in V}\sum_{k=1}^{K}\log(P(y_i|x_i,z_i=k,\theta))\sum_{z\setminus z_i} q(z\setminus z_i,z_i=k) \\ &-\lambda\sum_{(i,j)\in\mathcal{E}}\sum_{k=1}^{K}\left((b_{ik}-b_{jk})^2+\|w_{ik}-w_{jk}\|_2^2\right)\sum_{z\setminus\{z_i,z_j\}} q(z\setminus\{z_i,z_j\},z_i=z_j=k) \\ &= \sum_{i\in V}\sum_{k=1}^{K} q(z_i=k)\log P(y_i|x_i,z_i=k,\theta) \\ &-\lambda\sum_{(i,j)\in\mathcal{E}}\sum_{k=1}^{K}\left((b_{ik}-b_{jk})^2+\|w_{ik}-w_{jk}\|_2^2\right)q(z_i=z_j=k). \end{aligned}$$

where we already have joint posterior $q(z)$ calculated in the E-step, and we have to get the node marginal posterior distribution $q(z_i=k)$ and edge marginal posterior distribution $q(z_i=z_j=k)$. Based on equation (4), $q(z_i=k)$ is defined as below:

$$q(z_i=k)=P(z_i=k|y_i,x_i;\theta)=\frac{P(y_i|x_i,z_i=k;\theta)P(z_i=k;\theta)}{\sum_{s=1}^{K}P(y_i|x_i,z_i=s;\theta)P(z_i=s;\theta)}.$$

Similarly, based on equation (4), $q(z_i = z_j = k)$ is defined as below:

$$q(z_i = z_j = k) = P(z_i = k, z_j = k | y_i, y_j, x_i, x_j; \theta) = \\ \frac{P(y_i, y_j | x_i, x_j, z_i = k, z_j = k; \theta) P(z_i = k, z_j = k; \theta)}{\sum_{m=1}^{K} \sum_{n=1}^{K} P(y_i, y_j | x_i, x_j, z_i = m, z_j = n; \theta) P(z_i = m, z_j = n; \theta)}.$$

Notice that to compute the above probabilities it is required to know the marginal $P(z_i = s; \theta)$, which is expensive to compute since it requires marginalizing $P(z; \theta)$ which involves an exponential number of calculations. In the next section, we discuss how to compute the marginal $P(z_i = s; \theta)$ approximately.

4.5.4 Approximate Inference for the E step

The exact inference for our model is hard, but we can have approximate inference through sampling, to be more exact Gibbs sampling in our model. Our objective is to approximate the marginal distribution $P(z_i = s; \theta)\ \forall i$ by sampling from this distribution using the Gibbs sampling algorithm. Below we present the Gibbs sampling algorithm and the definition of the conditional probabilities that are required for sampling. After running Algorithm 4 we

Algorithm 4 Gibbs sampling for $P(z_i; \theta)\ \forall i$

```
Inputs:
    Parameters $w$ and $b$
Initialize:
    $z^{(0)} \sim q(z)$
for iteration $i = 1, 2, \ldots$ do
    $z_1^{(i)} \sim P(z_1 | z_{[N]\backslash 1}^{(i)})$
    $\vdots$
    $z_N^{(i)} \sim P(z_N | z_{[N]\backslash N}^{(i)})$
end for
```

obtain samples for each variable $z_i\ \forall i$. We can use these to form a histogram and use this distribution as an approximation to the marginal $P(z_i; \theta)$. Having this distribution we can perform inference and compute $P(z_i = k | y_i, x_i; \theta)$.

To implement Algorithm 4 we need formulas for the conditional probability $P(z_i = k | z_{[N]\backslash i})$. When there are K classes for the latent variables, the conditional probability is given by

$$\psi(i, j, k) = \left((b_{ik} - b_{jk})^2 + \|w_{ik} - w_{jk}\|_2^2 \right) \mathbf{1}(z_j = k)$$

and

$$P(z_i = k | z_{[N]\backslash i}) = \frac{\exp\left(-\lambda \sum_{j \in N(i)} \psi(i, j, k) \right)}{\sum_{\tilde{k}}^{K} \exp\left(-\lambda \sum_{j \in N(i)} \psi(i, j, \tilde{k}) \right)} \tag{4.10}$$

To perform the M-step we also need to know the edge marginal posterior $P(z_i = k, z_j = k|y_i, y_j, x_i, x_j; \theta)$ which in turn requires knowing the marginal $P(z_i = k, z_j = k; \theta)$. We compute the latter approximately by using the samples obtained by the Gibbs sampling Algorithm 4.

Chapter 5

Connected Population Synthesis

5.1 Introduction

Understanding how urban systems operate and evolve through modeling individuals' daily urban activities has been a major focus of transportation planners, urban planners, and geographers. To address operational needs in planning and policy decision making, reliable agent-based land use and urban transportation micro-simulation frameworks such as TRANUS [26], UrbanSIM [118], ILUTE [103], MATSim [6], MEPLAN [23] are becoming popular. Models implemented via micro-simulations require detailed information on each of the agents that represent the population in the region of study. Traditional ways of obtaining this information include using community survey data, or travel surveys based on individual or households travel diaries. These datasets provide a rich set of features but are limited in sampling size, geographical scope, and frequency of updates. The reasons for the limited availability of such detailed disaggregated data to researchers range from the lack of technical means and resources for surveying to personal information protection requirements, related data security and privacy concerns.

This work is motivated by the lack of inter-personal communication networks in the traditional data collection methods. This deficiency limits the development of the next generation of models that would appropriately integrate social effects with spatial-temporal information to better capture the dynamics of urban systems. To enable agent-based simulation in the presence of social influence effects, a connected synthetic population is needed as input. However, current state-of-the-art population synthesis models fail to generate social networks information because household surveys do not include social information. Alternatively, the prevalence of mobile phones provides a new data source for generating social networks. Pervasive sensing by telecom companies and location-based service providers generates large-scale geolocated communication datasets in which timestamped locations of users are recorded whenever calls are placed or messages are sent. An example of such data is the network carrier mobile phone usage logs, such as Call Detail Record (CDR) data. While manual surveying techniques are limited in their ability to collect social network data on a

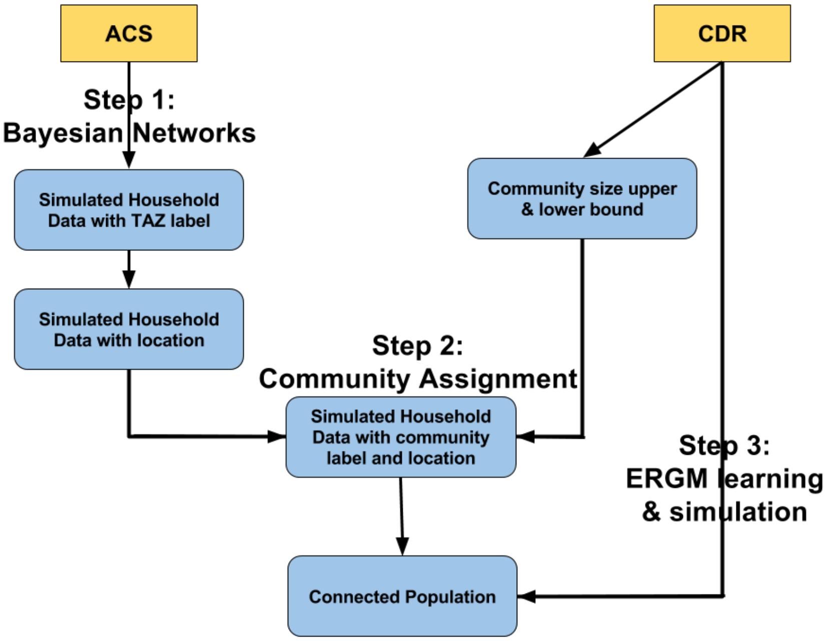

ACS
CDR
Step 1:
Bayesian Networks
Simulated Household Data with TAZ label
Community size upper & lower bound
Simulated Household Data with location
Step 2:
Community Assignment
Simulated Household Data with community label and location
Step 3:
ERGM learning & simulation
Connected Population

Figure 5.1: Population synthesis framework, ACS: American Community Survey Data, TAZ: Traffic Analysis Zone, CDR: Call Detail Record Data, ERGM: Exponential Random Graph Model

large scale, digital records of CDR provide an abundance information on spatial patterns of social networks [122].

In this paper, we propose an integrated methodology for incorporating various types of data into an integrated model that captures both social interactions and spatial patterns. It brings together several data components, reproducing the marginal and joint distributions of individuals and household level socio-economic characteristics, a geographical pattern of the observed community structure, and the statistics of the observed social networks.

The proposed population synthesis methodology includes the following steps, presented graphically in Figure 5.1. First, the household data is used to reproduce the socio-economic characteristics within the generated synthetic population. Next, the structure of the social network in the region is inferred from available network data and applied to connect the members of the generated households into a synthetic social network that follows the key structural properties of the observed one. The proposed methodology acknowledges the limitations of the data availability, as household and social network data are typically available from two separate sources with no implicit way to identify individuals present in both. Therefore, the sequence of methods that we introduce are aimed to model and reproduce key statistical characteristics of the connected population. The methods involved in the population synthesis are:

- **Step 1. Bayesian Networks:** Composition and socio-economic characteristics of the synthetic households are generated based on Bayesian network parameters estimated from a typical household survey data (such as the American Community Survey);

- **Step 2. Community allocation:** Integer programming problem of community assignment is solved with the Lagrangian Relaxation Method to enrich the simulated population with community membership;

- **Step 3. ERGM learning and simulation:** Parameters of an Exponential Random Graph Model are calibrated on the available social network data and applied to simulate social connections between the members of the synthetic population.

The rest of the paper is organized as follows. Section 5.2 presents the methods involved in each step, while comparing them to the state-of-the-art methods. The following Sections 5.3 and 5.4 illustrate the application of the methods on real data, simulating a connected population in the San Francisco Bay Area in California, US. Finally, Section 3.7 concludes the paper with a discussion of the achieved results and outlines the directions for future work.

5.2 Methods

In this section, we explain the methods explicitly adopted in each step as shown in Figure 5.1. We first introduce the Bayesian Networks estimated from American Community Survey (ACS) for simulating synthetic population in step 1. Then we use Lagrangian relaxation to solve the integer problem formulated for community assignment in step 2 with parameters learned from Call Detail Records (CDR). In the final step, we explain how the Exponential Random Graph Models (ERGM) learning is applied to social network simulation.

5.2.1 Step 1: Generating Synthetic Population

In this section, we explain how to use the state-of-the-art techniques to generate synthetic population, which is used as input for step 2 and step 3. The problem of this step can be regarded as the canonical synthetic population problem.

5.2.1.1 Related Work

Traditional population synthesis methods encompass two main directions: (1) Iterative Proportional Fitting (IPF) related models that focus on fitting a contingency table constructed from the micro samples to satisfy marginal distribution constraints from aggregated census data, along with the extensions of IPF models that aim to satisfy marginal distribution constraints of both individuals and households characteristics [132, 15, 143]; (2) statistical models that use MCMC (Monte Carlo Markov Chain) to sample a vector of socio-economic characteristics of each individual sequentially so that it captures the observed conditional relationships between the variables [38].

As a method of replicating existing sample data, IPF-related methods are sensitive to data quality and sample size. MCMC-based approach, however, also suffers from the drawback that it is hard to obtain the specified conditional distributions especially when one deals with many variables of interest. As pointed out by Sun et al. [115], Bayesian network approach is powerful in characterizing the underlying joint distribution, outperforming IPF and MCMC, and avoiding over-fitting the data.

5.2.1.2 Proposed Method: Bayesian Networks

Based on the vast amount of literature on population synthesis, we choose to follow the Bayesian networks approach similar to the approach in [115]. The Bayesian network approach is a generic formalism aiming at modeling the joint distribution of X from data. It consists of two main steps: (1) structure learning to define the Bayesian network structure $\mathcal{G}$ that describes the conditional independence of the random variables, and (2) parameter learning to learn a conditional distribution of random variables given this fixed directed acyclic graph (DAG) structure $\mathcal{G}$.

In the context of population synthesis, the objective of this method is to infer the multivariate probability distribution $P(X)$ of socio-economic parameters of households based on observed data. Socio-economic parameters $X \in R^d$ and $X_1, X_2, \cdots, X_d$ are typically composed of d discrete random variables representing the available information on both households and individuals, which was collected via surveying, and regarded as the complete set of observations $D = \{(x_1^t, x_2^t, \cdots, x_d^t), t \in [1, n]\}$ where $(x_1^t, x_2^t, \cdots, x_d^t)$ is one realization of X.

5.2.1.3 Structural learning and parameter estimation

The Bayesian network structure learning algorithms can be grouped into two categories: (1) constrained-based algorithms where dependencies are set using domain knowledge, and the resulting models are interpreted as causal models; (2) score-based algorithms where we select the graph structure $\mathcal{G}$ that results in the highest score following some accepted scoring criterion. In the present work, we use the algorithms from the second category.

For each graph structure $\mathcal{G}$, we factorize the joint distribution $P(X)$ as the product of conditional distributions $P(X_i|X_{\pi_i})$ where X_{π_i} represents the parent nodes of X_i given graph $\mathcal{G}$.

$$P(X; \mathcal{G}) = \prod_{i=1}^{d} P(X_i|X_{\pi_i}) \tag{5.1}$$

Then we can rewrite the log likelihood of the data given model parameters θ and the graph structure $\mathcal{G}$ in question as:

$$\begin{aligned} l(D;\theta,\mathcal{G}) &= \sum_{t=1}^{n} \log(P(x_1^t, ..., x_d^t; \theta, \mathcal{G})) \\ &= \sum_{t=1}^{n} \sum_{i=1}^{d} \log(P(x_i^t | x_{\pi_i}^t; \theta)), \end{aligned} \tag{5.2}$$

where $d = \dim(\mathcal{G})$ is the number of parameters (a table of conditional probabilities in case of the discrete variables), and n is the number of observations in D. Based on the definition of log likelihood, one can get the probability parameters $\hat{\theta}$ based on maximum likelihood estimation. One can then consider the maximum value of log likelihood for the graph structure $\mathcal{G}$, given estimated parameters $\hat{\theta}$ as $l(D;\hat{\theta},\mathcal{G})$. Each candidate Bayesian network can be assigned a score, and a final model can be selected based on the graph structure that results in the highest score. There are different network scores that can be used, among which the BIC (Bayesian information criterion) and AIC (Akaike information criterion)

$$BIC(\mathcal{G}) = l(D;\hat{\theta},\mathcal{G}) - \frac{d}{2}\log(n), \tag{5.3}$$

$$AIC(\mathcal{G}) = l(D;\hat{\theta},\mathcal{G}) - d, \tag{5.4}$$

are the most common choices. It is impossible to search over all possible graph structures $\mathcal{G}$ since the number of candidate graph structures increases super-exponentially with the number of variables [100]. The admissible set of structures can be defined using domain knowledge, and the selection process then can be optimized with some heuristic algorithms such as hill-climbing or tabu search. In this work, we followed [115] to use tabu search algorithm and AIC score for Bayesian networks structural learning. We further refer the reader to [54, 55, 19] for a detailed survey on Bayesian networks.

Parameters of a specific Bayesian network model are estimated for every type of a household, following a descriptive classification accepted in a region of study. Household tables are produced with an IPF method to match the marginal distribution of household types within the total population. Population synthesis includes sampling N entries from the $P(X)$, specific to the household composition dependencies captured by the structure of the graph G for the given type of the household. It results in a complete set of N individuals within H households with detailed socio-economic characteristics. Section 5.4.1 below illustrates the use of the method in the context of typically available survey data.

5.2.2 Step 2: Community Assignment

As justified in the introduction, community membership is one of the key features that we would like to reproduce in the connected population synthesis. We consider community to be defined as a group of individuals that possess stronger ties within the group as compared to the connections emanating to the outside of the group. Within communities, homophily pattern [78, 21, 109] are often observed. Spatial proximity is one of the most important homophily patterns for spatial communities. For example, urban sociology studies have identified communities based on neighborhoods and social activities [48].

Therefore, we acknowledge spatial proximity as a primary property that needs to be included in the community assignment step of population synthesis. Here spatial proximity means members within the same community are spatially not too far from each other, but not spatially clustered. For instance, people in the same office building are not necessarily in the same community. Therefore, geographically clustering algorithm cannot be directly used here. Besides, as observed from real-world data, most communities have a reasonable size bound, so we also need to constrain the community size. Considering the spatial proximity and community size, we formulate an integer optimization problem to solve the prescriptive assignment of community membership, which is then taken into account in the actual social networks simulation step using the model described in Section 5.2.3.

5.2.2.1 Formulation

To assign individual to communities, we formulate the community assignment as an integer programming problem. Without loss of generality, we consider spatial locations to be known at the resolution of a spatial zone according to the system defined by a population census. A random location within a zone can be assigned to each household, or a location following the exact address can be specified if available from a real estate or a cadaster dataset.

Assume a synthetic population of the total of N people is grouped into a total of K communities. Suppose $F \in R^{N \times N}$ is the feature distance matrix between individuals. The distance can be defined based on the vector of socio-economic characteristics to capture homophily relationships. The distance can also be extended (or replaced) with a similarity measure available from the data in hand. An appropriate model for F is therefore specifically determined by the application and is governed by the type of the simulation that the synthetic population is intended to serve. We provide an example in the experimental section below.

We further introduce c_j and Z_{ij} as decision variables. For the convenience of the assignment formulation, we define $c_j = 1$ if the j^{th} person is the social center of his/her community. This representation does not carry a functional meaning but defines a convenient way to enumerate the groups. Similarly, Z is an assignment matrix, with $Z_{ij} = 1$ if the i^{th} person belongs to the community centered around c_j and $Z_{ij} = 0$ otherwise.

In the presented approach, we formulate an optimization problem for the community assignment procedure. Assume we first select K individuals serving as centroids of the communities, and then assign other individuals to these centroids. Under the assumption that members within the same community tend to have similar features defined and available in F, we relax this assumption to the one in which people tend to have similar features as their community centroids, justifying the re-assignment step that maximizes the objective function. An individual can only belong to one community. Finally, the size of each community is bounded by the lower and upper bound detected from data or otherwise specified. This set of

Table 5.1: Notation

Variable	Definition
F_{ij}	the feature distance between individual i and j
Z_{ij}	individual i belongs to the community whose center is j
m	the smallest size of the community
M	the largest size of the community
u_j	if j^{th} individual is the center of the community

assumptions corresponds to the optimization problem below for the community assignment step (refer to Table 5.1 for notation).

$$\begin{aligned} \underset{Z}{\text{minimize}} \quad & \sum_{i=1}^{N}\sum_{j=1}^{N} F_{ij} * Z_{ij} \\ \text{subject to} \quad & \sum_{j=1}^{N} Z_{ij} = 1, \forall i; \\ & \sum_{i=1}^{N} Z_{ij} \leq M * u_j, \ \forall j; \\ & \sum_{i=1}^{N} Z_{ij} \geq m * u_j, \ \forall j; \\ & u_j = 1 \ \text{ or } \ 0, \forall j; \\ & Z_{ij} = 1 \ \text{ or } \ 0, \forall i, j. \end{aligned} \tag{5.5}$$

5.2.2.2 Lagrangian Relaxation

To efficiently solve the assignment problem we relax the condition $\sum_{j=1}^{N} Z_{ij} = 1$, using the Lagrangian Relaxation Lower Bound (LLBP) method that results in an formulation defined in Equation 5.6:

$$\begin{aligned} \underset{Z}{\text{minimize}} \quad & \sum_{i=1}^{N}\sum_{j=1}^{N} (F_{ij} + \lambda_i) * Z_{ij} - \sum_{i=1}^{N} \lambda_i \\ \text{subject to} \quad & \sum_{i=1}^{N} Z_{ij} \leq M * u_j, \ \forall j \\ & \sum_{i=1}^{N} Z_{ij} \geq m * u_j, \ \forall j \\ & u_j = 1 \ \text{ or } \ 0, \ \forall j \\ & Z_{ij} = 1 \ \text{ or } \ 0, \ \forall i, j \end{aligned} \tag{5.6}$$

This problem is separable, so we write the sub-problem as j-problem:

$$\begin{aligned}
\underset{Z}{\text{minimize}} \quad & \sum_{i=1}^{N}(F_{ij}+\lambda_i)*Z_{ij}-\lambda_i \\
\text{subject to} \quad & \sum_{i=1}^{N} Z_{ij} \le M*u_j \\
& \sum_{i=1}^{N} Z_{ij} \ge m*u_j \\
& u_j = 1 \;\text{ or }\; 0 \\
& Z_{ij} = 1 \;\text{ or }\; 0\;, \forall i
\end{aligned}$$

This problem can be solved numerically using Algorithm 5. For the j-problem, we can solve it following the steps below.

1. If $u_j = 1$, we pick at least m and at most M units of Z_{ij} among i to be 1, that is, if u_j is a community center, then at least m people and at most M people should belong to this community. We compute $F_{ij} + \lambda_i$ for each i and rank them in an ascending order.

2. We denote N_c as the number of negative values for $F_{ij} + \lambda_i$ with all i and fixed j. If $N_c \le m$, we sum the first m smallest coefficients. If $m < N_c \le M$, we sum the first N_c smallest coefficients. If $N_c > M$, we sum the first M coefficients. We define S as this sum. Since the goal is to minimize the objective function, we set $u_j = 1$ if $S < 0$ and set $u_j = 0$ otherwise.

3. For any λ, what we calculate for Lagrangian relaxation is a lower bound, which is denoted as LB. To update λ, we also need to compute the upper bound, which is denoted as UB. Let $L = \{i | \sum_j Z_{ij} = 0\}$, which denotes the set of points that are not assigned yet. We randomly pick $\lceil |L|/M \rceil$ points from the set L as centers C. Then we set each point $i \in L$ to a center $j \in C$ until each community has m people according to the ascending order of $F_{ij} + \lambda_i$ for each i. After that, we continue to add the unassigned points $i \in L$ to center $j \in C$ until the number of points belonging to j reaches M. When all members are allocated, we find a feasible solution and thus an upper bound for this problem. Lagrange multipliers λ_i are then updated as follows: $\lambda_i^{t+1} = \lambda_i^t + \Delta^t(\sum_{j=1}^{N} Z_{ij} - 1)$ with the step size $\Delta^t = \frac{\alpha(UB-LB)}{\sum_i(\sum_j Z_{ij}-1)^2}$.

The output of this algorithm is the community assignment matrix Z and membership labels u that match the required spatial structure and sizes of the observed communities. It will be used in social network simulation step as described below.

Algorithm 5 Community Assignment Lagrangian Relaxation Algorithm

```
Initialize: $\lambda = 0, \alpha = 0.01$

while $\frac{UB-LB}{LB} > \epsilon$ do Solve LLBP given $\lambda$ values to get $LB, c_j, Z_{ij}$

Find a feasible solution to get $UB$
Calculate the subgradients $G_i = \sum_{j=1}^{N} Z_{ij} - 1$
Calculate the step size $\Delta^t = \frac{\alpha(UB-LB)}{\sum_{i=1}^{n} G_i^2}$
Update $\lambda_i = \lambda_i + \Delta^t G_i$
```

5.2.3 Step 3: ERGM learning and social network simulation

Various methods and software have been proposed for social network simulation, such as Forest Fire model [70], Exponential Random Graphs Models [114], Markov Random Graphs Model [41], and YANG (Yet Another Network Generator) [117], etc. In this paper, the objective of the final step is to generate social connections for the simulated population data enriched with community assignment and household locations. For this purpose, we adopt the Exponential Random Graph Model (ERGM) to learn the parameters from available social network data and use the learned parameters to create the social connections between the synthetic population members. This step concludes with the generation of the complete connected population enabling transportation simulation with social extensions.

5.2.3.1 Methodology review of ERGM models

Among the large amount of literature on the existing models of social networks, one cornerstone approach is the exponential random graph models. It is a set of models that assume the probability of the existence of certain graph structure with the corresponding adjacency matrix a as belonging to an exponential family

$$Pr(A = a|\theta) = \exp(\theta^T s(a) - \psi(\theta)), \tag{5.7}$$

where θ is the vector of unknown parameters, $s(a)$ is the vector of sufficient statistics computed on the adjacency matrix a, such as the counts of subgraphs like triangles and k-stars, and ψ is the normalizing constant. It is therefore focused on deriving probabilistic models of graphs that match the statistics of the structural properties of the observed networks.

The work on Exponential Random Graphs Models can be classified into two groups concerning the statistical independence or dependence of links.

Type I ERGM: Links are independent. The sufficient statistics in this case is the total number of links. In the simplest case introduced by Erdós et al. [36], one assumes all network edges are random variables that follow the same distribution. The model can be parameterized with more variables when more information on the node properties becomes available. For example, the classical stochastic block model [114, 88] incorporates community assignment. It

assumes that the probability of the existence of an edge between two nodes within the same community and the probability of the existence of an edge between two nodes from different communities are different. Some other models [127] use ERGM with the assumption that the edge probability between any two nodes depends on the characteristics difference. Variables such as age, gender [53], as well as spatial distance [127] can be considered as factors defining the edge probabilities. Such models incorporate the homophily assumptions and are used extensively in social sciences [99, 113, 98].
Type II ERGM: Links are dependent. Link independence, resulting in Bernoulli and dyadic dependence structures, are unrealistic assumptions in many circumstances, both empirically and theoretically. To address this shortcoming, models which add the dependence of links were proposed. The classical model is a Markov Random Graphs Model [41], where the number of triangles and number of k-stars are used as sufficient statistics, so that two edges are supposed to be conditionally dependent, given the values of all other ties. A major difficulty in Type II ERGM model inference is how to evaluate $\psi(\theta)$, as there is no feasible analytic method for approximating $\psi(\theta)$ for large networks [96]. Various Monte Carlo schemes have been proposed to approximate $\psi(\theta)$, but two fundamental difficulties of this type of ERGM model still remain. First, the estimation result using Markov Random Graph Models is not robust, and Monte Carlo methods converge to ERGM model with link independence within mixing time. Second, Markov Random Graph Model is not scalable enough to handle large network inputs, which means that the parameter estimation in practice will be computationally slow, if at all possible.

5.2.3.2 Community-distance ERGM

Given the limitations of conventional ERGMs mentioned above, we propose the "community-distance" ERGM model where we make the following assumptions: (1) links are independent; (2) people who are characteristically close to each other are more likely to be connected; (3) people in the same community are more likely to get connected with each other.

We, therefore, aim to infer the parameters θ based on observation of adjacency matrix A, characteristics distance matrix F and community labels C from available social networking data. The learned parameter θ is then used to generate links for the simulated population, given the characteristics distance matrix F and community assignment C where $C_i = j$ indicates that the community label of individual i is j.

The probability of the social graph in our model is:

$$Pr(A = a|X, F) = \frac{1}{\mathcal{K}} \exp(\begin{bmatrix}\theta_1 & \theta_2 & \theta_3\end{bmatrix} \begin{bmatrix} \sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij} \\ \sum_{\{(i,j):C_i=C_j\}} A_{ij} \\ \sum_{i=1}^{N}\sum_{j=1}^{N} F_{ij}A_{ij} \end{bmatrix}) \tag{5.8}$$

where $\sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij}$ is twice the total number of edges, $\sum_{\{(i,j):C_i=C_j\}} A_{ij}$ represents twice the number of edges that connect two nodes belonging to the same community, and $\sum_{i=1}^{N}\sum_{j=1}^{N} F_{ij}A_{ij}$ represents twice the sum of the distances between pairs of nodes connected by an edge. To transform this problem into standard logistic regression problem, we introduce a random variable Y_{ij}, with $Y_{ij} = 1$ when $A_{ij} = 1$, and $Y_{ij} = -1$ when $A_{ij} = 0$. Model parameters can then be estimated based on maximum likelihood, given by

$$\begin{aligned} l(A = a; \theta) = \log(P(A = a|X, F)) &= \sum_{i=1}^{N}\sum_{j=1}^{N} \log(P(A_{ij} = a_{ij})) \\ = \sum_{i=1}^{N}\sum_{j=1}^{N} \log(P(Y_{ij} = y_{ij})) &= \sum_{i=1}^{N}\sum_{j=1}^{N} \log(\frac{1}{1 + e^{-y_{ij}s_{ij}}}) \end{aligned} \tag{5.9}$$

where

$$s_{ij} = \begin{cases} \theta_1 + \theta_2 + \theta_3 F_{ij}; & \text{if } C_i = C_j \\ \theta_1 + \theta_3 F_{ij}; & \text{otherwise.} \end{cases} \tag{5.10}$$

Unlike Type II ERGM models where links are dependent, this model specification does not require MCMC procedure to generate possible graph structures, and maximum likelihood estimation provides numerically stable and robust parameter estimation $\hat{\theta}$. The social network for the synthetic population A can then be simulated given the characteristics distance matrix F and community assignment C, with

$$\Pr(A_{ij} = 1) = \frac{1}{1 + e^{-\hat{s}_{ij}}}, \tag{5.11}$$

where

$$\hat{s}_{ij} = \begin{cases} \hat{\theta}_1 + \hat{\theta}_2 + \hat{\theta}_3 F_{ij}; & \text{if } C_i = C_j \\ \hat{\theta}_1 + \hat{\theta}_3 F_{ij}; & \text{otherwise.} \end{cases} \tag{5.12}$$

The following sections of the paper illustrate the use of the aforementioned methodology and describe the practicalities of the applications of methods to the particularities of available data.

5.3 Data

Model specifications for an application of the described methods depend on the availability of data. This section describes two typical data sources that are used to illustrate the synthesis of individuals for a given type of the households, and a corresponding social network of the required community structure for this synthetic population. Due to privacy protection regulations, it is unlikely that the two components of data will become readily available in a way enabling matching the users across two sources, motivating the development of the population synthesis methodology presented in this paper.

5.3.1 American Community Survey

The first data source we utilized in the experiments part of this paper is a typical household survey data, known as the Public Use Microdata Sample (PUMS) of the American Community Survey (ACS). It contains multiple socio-economic parameters of the household members for a sample of the households in the region. This sample size typically ranges from 1% to 10%, and is maintained by regional governments and agencies. Apart from this micro sample data set, the aggregated marginal distribution of the population totals at the block group and census tract levels are also provided.

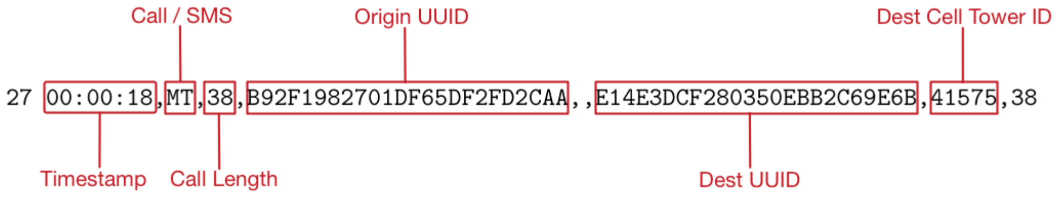
In the presented work, the region of study encompasses the San Francisco Bay Area: a 7,000 square-mi region spanning nine counties under the jurisdiction of the Association of Bay Area Governments. According to the updated information from Metropolitan Transportation Commission and the Association of Bay Area Governments, there are 7 million people residing in nine counties and 101 cities. There are in total 1588 census tracts and 72 Public Use Microdata Areas. The data available from the ACS database carries the records for 439,525 people from 132,311 households, which corresponds to a sampling rate of 6.1%.

By the ACS classification, households are divided into several different types based on the household size and structure. In this paper, we use 2-people households as an example. There are a total of 23895 2-people households in the PUMS of the study area.

5.3.2 Social Network Data

Social network data sources range from small-scale samples collected from research projects to massive repositories of users data kept by online social network companies and telecom operators. One source of the latter type is known as the Call Detail Records (CDR), which is a standardized format of call logs collected by the operators of cell phone networks. Below we give a brief introduction to CDRs data and then elaborate on how we construct the social networks from these records.

Collected by cellular network operators for billing purposes, CDR datasets contain several features that have helped fuel the burgeoning field of computational social science. Each record describes a communication event on the cellular network. It contains Universally Unique Identifiers (UUIDs) representing the anonymized calling (or texting) and receiving individuals, the time of initiation, its length (if it was not a text), and the unique identifiers of the cell towers at the outgoing and incoming locations (see Figure 5.2 for an example).



```
27 00:00:18,MT,38,B92F1982701DF65DF2FD2CAA,,E14E3DCF280350EBB2C69E6B,41575,38
```

Figure 5.2: A sample CDR(Call Detail Record) with highlighted data fields.

5.3.2.1 Data preparation for network generation

To construct a social network using CDR data, certain pre-processing operations are often necessary in order to filter out spurious calls, marketing calls, and other interactions not necessarily indicative of social contact [10, 67, 89, 90]. Regarding the construction of the CDR network, there are multiple ways to represent edges between two individuals. Key choices depend on the desired network representation to be weighted/unweighted and directed/undirected. For the purpose of population synthesis, we choose to construct the social network as unweighted and undirected. We add an edge between individuals i and j if individual i called/received a call or send/received a message from individual j, and it has been reciprocated within the time span of the dataset. In the sample available to us from a telecom operator, the resulting network representation consists of 1,321,765 edges and 343,299 nodes. This way of constructing the CDR network is based on Section 2.1 of [9] and Section 2.1.1 of [77]. No personally identifiable information is available due to privacy protection, making it impossible to match the users to the community survey sample, or to assign the users with the exact set of the socio-economic parameters. Instead, the key objective of the proposed methodology is to infer the structural properties of the social networks in order to reproduce them in its synthetic version.

5.3.2.2 Home and work locations

The first characteristic that is required for social network synthesis is the spatial spread of the detected communities. It defines the decay of the edge probability with the distance between nodes as well as the geographical boundaries between the communities that are known to be different from the administrative or other artificially defined divisions [67]. A set of "anchor" locations such as home and work are required in order to define characteristic spatial structure of communities that we use in the assignment algorithm described above in Section 5.2.2. There are mainly two popular ways for home and work detection: first, a Gaussian mixture model is adopted to model locations centered around home and work [20]; second, "home" is defined as the location where the user spends more than 50% of her/his time during night hours. Similarly, "work" is defined as the location where the user spends more than 50% of her/his time during day hours [66, 134, 93]. In this paper, we adopt the similar approach as in [134], and we define 6pm-8am as night hours and 8am-6pm as day hours.

5.3.2.3 Community Detection

The second characteristic of interest is the community structure of the social network. To obtain the largest community size M and the smallest community size m for the improved two-stage Lagrangian relaxation method, we need to conduct community detection on Call Detail Records social networks to obtain these features.

Within the variety of state-of-the-art methods of community detection, two main approaches are modularity-based methods [87] and conductance-based methods [61], with modularity and conductance correspondingly serving as the criteria for community detection. Conductance is a local metric, whereas modularity is a global metric. The smaller the conductance is, or the bigger the modularity is, the better the community detection is.

The majority of studies involving CDRs for community detection use modularity due to their focus on intra-community size and homogeneity. For example, a common approach is to use the well-established Louvain method [10], as exemplified by [122, 57], with extensions such as fast hierarchical agglomerative clustering based on the modularity metric [42, 110]. However, modularity-based methods fail to capture small-scale community patterns [71] typical for small social groups at the scale of households that we are interested in. Thus we propose to use conductance-based local community detection methods [61], particularly the local graph clustering [40, 112], for finding realistic upper and lower bound for community size required for the assignment step described in Section 5.2.2. We refer the reader to [39, 4] for detailed surveys of conductance-based methods.

5.4 Experiments

We provide an illustrative experiment by generating a synthetic connected population within the San Francisco Bay Area, California. The experiment was run on a MacBook with 2.5 GHz Intel Core i7 processor and 16 GB memory. The results at each step of the methodology are described in detail for the county of Napa, which consists of 108 census tracts and a total of 48,876 households with 131,556 residents according to the US Census Bureau.

5.4.1 Household synthesis

A specific Bayesian network model has to be produced for each household category, calibrated from micro-census sample and applied to synthesize the required number of households of each type following the available aggregate numbers of household types in the area of interest.

For simplicity of the illustration, out of the total set of observations with 500 variables contained in the PUMS, we restrict the example to 23,895 2-people households. The latter consists of the head of the household (the householder) and a spouse or a domestic partner. For the set of socio-economic parameters, we choose the variables that are of interest to urban planners and transportation researchers, which are: SEX (sex of householder), SEX_S (sex of householder spouse), AGE_index (age level of householder), AGE_S_index (age level of householder spouse), PINCP_index (income level of householder), PINCP_S_index (income

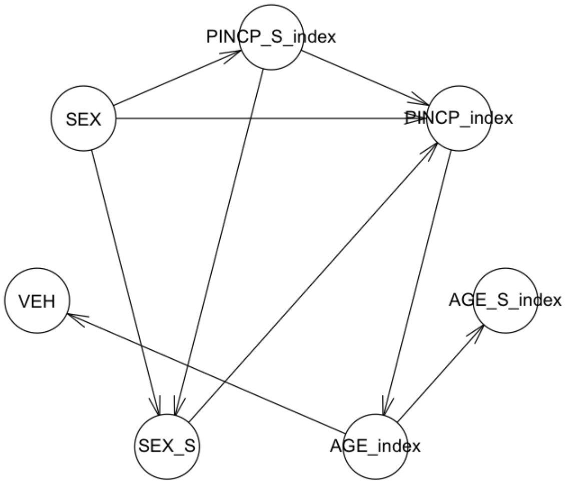


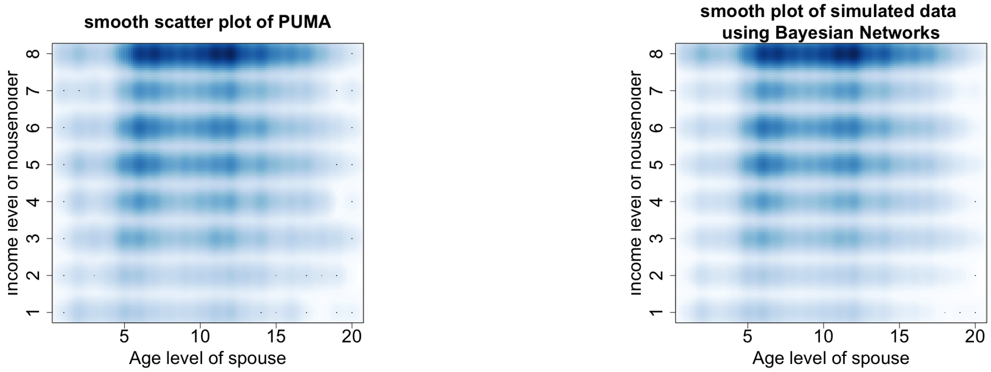
Figure 5.3: The learned structure of the Bayesian network for 2-people households.

level of householder spouse), and VEH (number of vehicles owned by this household). We factorized raw input data from Public Use Microdata Sample (PUMS) to categorical variables based on the corresponding ACS age and income level bins so that it can be matched with aggregated marginal distributions in full population synthesis.

5.4.1.1 Bayesian Network Structure and Parameter Learning

Within each type of a household, the dependency structure between the chosen variables needs to be defined. This problem is known as the structure learning. By applying the tabu search methods [115], we realized that the estimated model structure is not robust when we bootstrap or use multiple data subsets. This behavior was not reported in [115]. To overcome the undesirable outcomes and constrain the dependencies based on the domain knowledge and common-sense relationships, we defined a "whitelist", which defines a set of relationships in structural learning procedure that must be preserved and are guaranteed to be present in the final graph. The whitelist used in our experiments included the gender/income dependence, which means the arrow, SEX → PINCP_index, is whitelisted. Figure 5.3 presents an estimated model structure for 2-people households.

Parameter learning for the fixed structure of the Bayesian network is straightforward. We use an implementation of bnlearn R package [108] for the parameter learning and sample



(a) PUMS data

(b) Bayesian Network Model Simulated Data

Figure 5.4: Joint distribution of the household head income and the age of the spouse/partner in the 2-people households.

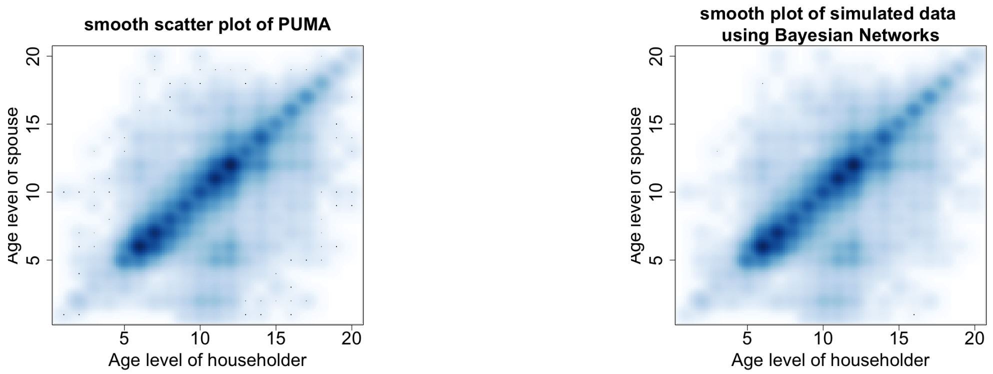
from the final model for population synthesis.

5.4.1.2 Metrics Evaluation

To keep the presentation concise, we report on the representative results of the parameter estimation without providing a full set of tables for conditional distributions between pairs of variables (shown in Figure 5.3). Figure 5.4 shows the joint distribution of owner income and spouse age of PUMA and simulated data from Bayesian network, and Figure 5.5 shows the joint distribution of owner age and spouse age of PUMA and simulated data from Bayesian network. It illustrates that the Bayesian network satisfies the joint distribution of variables. One can clearly observe that the pairs of variables are not independent, and the dependency structures between the variables of household members are preserved.
To quantitatively access the performance of the synthetic population simulation, we measure the Kullback–Leibler divergence (KL divergence) (5.13), which is used to measure the difference between two probability distributions.

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{5.13}$$

However, in practice, it is very difficult to compute the KL divergence between the joint distribution of simulated data and input survey data due to the large categorical tuple space. As a trade-off, we calculate the KL divergence of the joint distributions in the Figure 5.4 and the joint distributions in the Figure 5.5 as examples of the overall simulation performance. The KL divergence of joint distribution of owner income and spouse age for simulated data and survey data data, as shown in Figure 5.4, is 0.008, and the KL divergence joint distribution of owner age and spouse age for simulated data and survey data, as shown in Figure 5.4, is $6.4e^{-5}$.



(a) PUMS data

(b) Bayesian Network Model Simulated Data

Figure 5.5: Joint distribution of the age of the head of the household and the spouse/partner in the 2-people households.

5.4.1.3 Synthesis with Bayesian networks and marginal distributions

Marginal distribution or joint distribution of two or more variables, which comes from aggregated census data, can be used as a metric to control the quality of sampled populations from the inferred Bayesian network [115]. However, it is instead desirable to match the observed marginal distributions precisely. Although one can access all aggregated feature variables for different geographical zones, in Bayesian network models, it is only possible to precisely match the distribution of the mother node. That is because based on the conditional distribution theorem, once the distribution of mother nodes and the conditional probability table are known, the distribution of child nodes are uniquely defined [54, 55, 19]. Thus it is impossible to satisfy the conditional distributions while satisfying the marginal distributions of all variables at the same time. A theoretically justified method for the latter is an open research problem. An acceptable practical strategy is to keep the Bayesian network as simple as possible, only capturing the dependencies between variables with strong statistical significance in addition to the white-listed ones. In our application, we observed the best results when we maintained the marginal distribution of the sex of the head of a household (mother node "sex" in Figure 5.3) according to the gender distribution from the aggregated census data.

5.4.2 Community Assignment

In this step, we need to obtain the necessary parameters for community assignment, including the F_{ij} matrix for the simulated population of the households, as well as the lower bound m and upper bound M of community size, based on the community detection results on available social network data set.

5.4.2.1 Community detection result

Structural properties of the social network in the presented application were detected from the available CDR data described in Section 5.3. We use the parallel local graph clustering method [112] implemented in Ligra [111] for the community detection. Because 77% of the communities have 30-50 individuals, we set the size range for m and M to be 30 and 50, respectively. Then, home and work locations were detected with the algorithm described above. The detected communities were found to be clustered geographically, while preserving the different characteristic spatial scale of their spread within the high and low population density areas.

We then aim at reproducing the spatial proximity and geographical boundaries within the detected community and the synthetic population, using the pairwise distances between the locations of the synthetic households as elements of F_{ij}. Having introduced a generic method of assignment communities to reproduce the observed spatial structure, we leave the detailed study on modeling F_{ij} and the relative importance of social versus spatial factors to further research.

5.4.2.2 Community Assignment Experiments

With derived values of m, M and F_{ij} we then perform community assignment. Since the large-scale assignment problem is quite time-consuming to solve, we implement the community assignment using CPLEX solver [25] and parallel computing paradigm known as Message Passing Interface (MPI) [46]. We randomly partition the input data into K groups, each with 300 individuals, and use CPLEX solver for each sub-problem, coordinated by MPI. We have controlled the running time of CPLEX by setting the relative MPI gap tolerance to 20%. Because we are not interested in finding the exact minimum value of the original problem, we can tolerate an approximate assignment. CPLEX with MPI was found to provide a sufficient small objective value while satisfying all the constraints. We found that it is also much faster than solving the original large mixed integer program.

5.4.2.3 Metrics Evaluation

Figure 5.6 illustrates the assigned communities color-coded on a map overlay, illustrating the spatial spread of the communities. We can notice the characteristic structure of a more spatially homogeneous communities in the rural areas and higher overlapping ones in the suburban zones, also with the county capital town as observed in previous studies [69]. It shows that our community assignment algorithm guarantees that individuals in a community geographically close to each other but not at the same location. If we apply other geographical clustering algorithms, like K-means or DBSCAN for the community assignment procedure, members of each community will be geographically clustered, which is not what Call Detail Records data (CDR) implies.

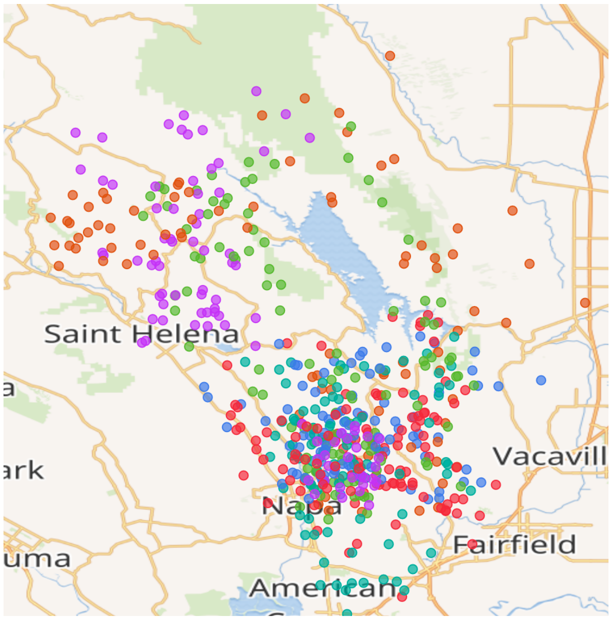


Figure 5.6: Mapping Assigned Communities

5.4.3 Step 3: ERGM learning & simulation

In this step, we apply the community-distance ERGM model implemented using CVX solver. We learn model parameters based on a subsample of a CDR network, and generate links for synthetic individuals. We then overlay the communities on a map and explore the network statistics of the simulated connected population.

5.4.3.1 ERGM learning

In our experiments we tried to adopt the type II ERGM model with dependent links, using the current state-of-the-art R ERGM implementations: (1) block exponential random graph model from "blkergm" package [129], and (2) exponential random graph model with local dependence [107] from "hergm" model [106]. However, both models were found to be not scalable to efficiently process the sampled CDR networks. Since there are no existing scalable parameter learning methods for Markov Random Graph Model that can efficiently handle large networks [105], we propose to use a model where edges are independent. However, even with the state-of-the-art R ergm packages [59] for the type I ERGM models, the estimation result is still not satisfiable because the implementation involves Monte Carlo MLE estimates. Here we report on the results achieved by maximizing the convex maximum likelihood function of the community-distance ERGM model. We implemented it using CVXPY [31].

Based on Equations (5.11)(5.12) and estimation result of Table 5.2, we observe that the probability of a connection decreases with distance. The probability of two individuals getting connected when they are in the same group is bigger than when they are not in the same group, since we have $\frac{1}{1+e^{-(\theta_1+\theta_3*D_{ij}+\theta_2)}} > \frac{1}{1+e^{-(\theta_1+\theta_3*D_{ij})}}$ because $\theta_2 > 0$ (check [16] for more

Table 5.2: Estimation result of community-distance ERGM

Parameters	Estimate
θ_1	-0.8803
θ_2	0.1182
θ_3	-14.5637

details), and it corresponds to the increasing trend of sigmoid function. This indicates that people who are in the same community are more likely to get connected.

5.4.3.2 ERGM simulation and Metrics Evaluation

With parameters learned from community-distance ERGM learning step, we can simulate the social connections for the simulated households. We illustrate and report on the properties of the simulated social networks. We report the performance of ERGM simulation in the following three metrics:
(1) Degree Distribution Figure 5.7 shows the degree distribution in log scale. Since the straight line fits the points quite well where $R^2 = 0.85$, it shows strong evidence of power law distribution.
(2) Distance Effect Figure 5.8 shows a geographical visualization of simulated social networks. It shows that people who live closer have higher probability to be connected.
(3) Community Membership Effect We also attempt to illustrate that people who are in the same community are more likely to get connected (given the same distance). We provide Figure 5.9 (right), which is a visualization of 10 communities within the simulated social network. As one can see from Figure 5.9 (right), there are people living far away that are connected, but they are most likely to be in the same community in accordance with the parameter estimation results. Note that there are some isolated nodes in Figure 5.9 (right) for two reasons: (1) this graph is a subgraph sampled from the simulated social networks, and some nodes are connected with ones which are not in this subgraph; (2) ERGM is a probabilistic model, and it is possible that the probability of one given node connected with each other node is small.

5.5 Potential Applications

Large volumes of call detail records (CDR) from mobile phones have been adopted in multiple transportation modeling framework [62, 74, 134]. These frameworks have similar data processing procedures: (1) generating synthetic population for a region, (2) producing synthetic travel plans for the population in the region by sampling models trained on CDR data, (3) using travel plans as inputs to an agent-based microscopic traffic simulator [6].

Since the social network property of call detail record (CDR) is not exploited, we can further use this information to obtain the statistics of the observed call detail record (CDR). Combined

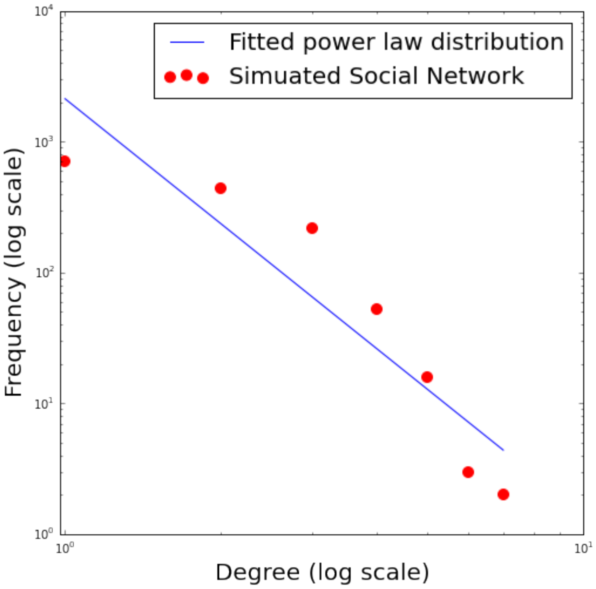


Figure 5.7: Degree Distribution plot of simulated social networks

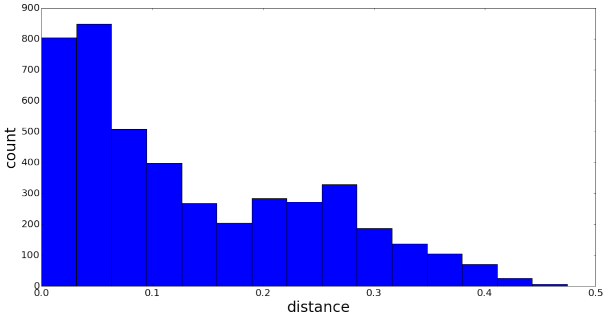


Figure 5.8: Histogram plot of distance between two individuals connected by an edge

with the marginal and joint distributions of individual and household level socio-economic characteristics, and a geographical pattern of the observed community structure, (1) social activity prediction model can be adopted to generate synthetic travel plans considering the effect of social connections; (2) social discrete choice model [138] can replace the multinomial discrete choice model in agent-based microscopic traffic simulator, to model the mode choice; (3) peer pressure effect [102] can be efficiently integrated into the agent-based microscopic traffic simulator.

5.6 Acknowledgement

This work was partially funded by a gift from AT&T. Support from the State of California Department of Transportation (CalTrans) through UCCONNECT faculty research grant program, agreement 65A0529, is also acknowledged. We thank Siyu Chen from Didi Chuxing

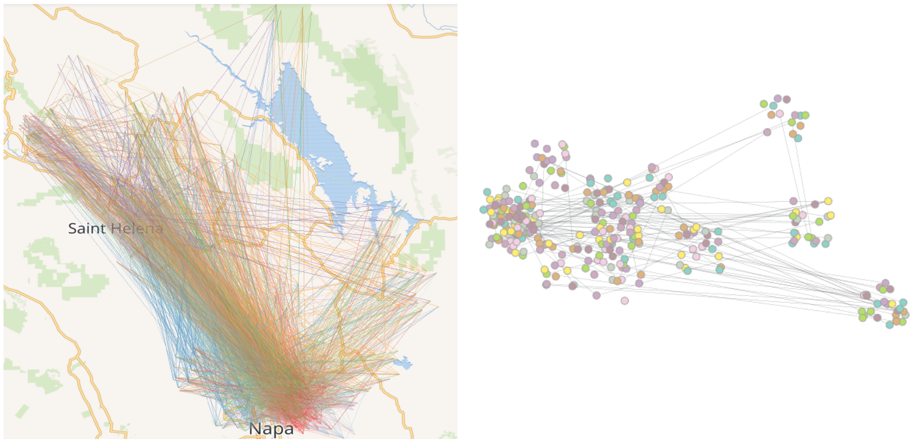


Figure 5.9: Social Networks visualization in a geographical space (left), and a subgraph for selected 10 communities in a network layout representation (right).

for assistance with data processing.

Chapter 6

Conclusion

6.1 Summary and conclusions

Understanding how friendship influences human behavior over time and space is the key topic of social-enabled urban data analytics. In this era of transformative mobility, this can help better design policies and investment strategies for managing existing urban infrastructure and forecasting future urban infrastructure planning. In this dissertation, we explored two research directions on social-enabled urban data analytics. First, we developed new machine learning models for social discrete choice model, bridging the gap between discrete choice modeling research and computer science research. Second, we developed a methodology framework for synthetic population synthesis using both small data and big data.

In the first part of the dissertation, we introduced social graph regularization ideas into discrete choice models for user choice modeling. We proposed local logistics graph regularization (LLGR) method and latent class graph regularization (LCGR) model. We developed scalable parameter estimation method for LLGR model on large graphs benefiting from recent advances in distributed optimization based on ADMM methods. Also, we have developed, implemented, and explored parameter estimation algorithms that allow parallel processing implementation for both E- and M-steps of the Monte Carlo Expectation Maximization (MCEM) algorithms for LCGR model. In experimental evaluation, we have focused on investigating the usefulness of the models in revealing and supporting the hypothesis in studies where not only predictive performance (that was found to be highly competitive) but also understanding social influence, is crucial. Our models can be directly applied to study social influence on revealed choices in large social graphs with rich node attributes. One challenge with extending our results is that such data are very rarely available in open access due to privacy issues.

In the second part of the dissertation, we developed an algorithmic framework to incorporate social information in population synthesizer.This proposed framework for connected population synthesis is applicable to cities or metropolitan regions where data availability allows for the estimation of the component models. The framework utilizes both traditional data sources such as household survey and census data (such as an ACS PUMS (public micro sample) and ACS aggregated census data in the context of the United States), and social network information that can be available for the region from cellular records or social media data. We implemented the proposed methods in code using state-of-the-art R packages [59, 108] and optimization toolboxes CPLEX [25] and CVX [31], applying an MPI parallel computing approach [46] to guarantee scalability. The code we developed in this work is available at `https://github.com/DanqingZ/CPS_TRC`.

A practical application of the proposed methods has demonstrated its usefulness. We presented an example illustrating the application of the approach to simulating a connected population for the Napa County, California, describing the modeling choices we made. The results have shown that the simulated connected population successfully captures a pattern from household survey data and transforms the observed community structure into a simulated population. We believe this framework presents a starting point for connected population synthesis research.

6.2 Future Work

The presented work has faced several limitations that we expect to address in future research. Particularly, we expect new developments to emerge along the lines of:

- Social Discrete Choice Models
  - Explore deep learning based models in the E step of MCEM to speed up the computation
  - Explore deep learning method to approximate the probability distribution instead of solving them explicitly.
- Connected Population Synthesis
  - Amending the Bayesian Networks modeling step with an advanced method that simultaneously allows: (1) fitting parameters of Step 1 by constructing a constrained optimization problem where the objective function is the data likelihood under a given network structure and meanwhile satisfying marginal distributions; (2) extending the current work to generate hierarchical structures of the social networks;
  - Generating more realistic home locations based on third-party real estate data;
  - Replacing ERGM with a more scalable social network generation method that can both handle large input networks and account for link dependence, particularly incorporating the effect of the households structure;

  - Thoroughly applying the synthetic population within a larger-scale agent-modeling framework.

- Other Research Direction in Social-enabled Urban Data Analytics

  - Introduce social information in analyzing human trajectory data. Develop new machine learning/deep learning models for social-enabled activity recognition and location prediction models.
  - Design robust social carpool mechanism to incorporate the social carpooling mode choice into the microsimulation system.

Bibliography

[1] J. Abernethy, O. Chapelle, and C. Castillo. "Graph regularization methods for web spam detection". In: *Mach. Learn.* 81.2 (2010), pp. 207–225.

[2] Fahad Alhasoun, May Alhazzani, Faisal Aleissa, Riyadh Alnasser, and Marta González. "City scale next place prediction from sparse data through similar strangers". In: *Proceedings of ACM KDD Workshop, Halifax, Canada.* 2017.

[3] May Alhazzani, Fahad Alhasoun, Zeyad Alawwad, and Marta C González. "Urban Attractors: Discovering Patterns in Regions of Attraction in Cities". In: *arXiv preprint arXiv:1701.08696* (2016).

[4] Reid Andersen, Fan Chung, and Kevin Lang. "Local graph partitioning using pagerank vectors". In: *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06).* IEEE. 2006, pp. 475–486.

[5] Jonn Axsen and Kenneth S Kurani. "Interpersonal influence within car buyers' social networks: applying five perspectives to plug-in hybrid vehicle drivers". In: *Environment and Planning A* 44.5 (2012), pp. 1047–1065.

[6] Michael Balmer, Marcel Rieser, Konrad Meister, David Charypar, Nicolas Lefebvre, Kai Nagel, and K Axhausen. "MATSim-T: Architecture and simulation times". In: *Multi-agent systems for traffic and transportation engineering* (2009), pp. 57–78.

[7] Mikhail Belkin, Irina Matveeva, and Partha Niyogi. "Regularization and semi-supervised learning on large graphs". In: *Conference on Learning Theory.* 2004.

[8] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples". In: *Journal of machine learning research* 7.Nov (2006), pp. 2399–2434.

[9] Vincent D Blondel, Adeline Decuyper, and Gautier Krings. "A survey of results on mobile phone datasets analysis". In: *EPJ Data Science* 4.1 (2015), p. 1.

[10] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. "Fast unfolding of communities in large networks". In: *Journal of statistical mechanics: theory and experiment* 2008.10 (2008), P10008.

[11] John Bongaarts and Susan Cotts Watkins. "Social interactions and contemporary fertility transitions". In: *Population and development review* (1996), pp. 639–682.

[12] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. “Distributed optimization and statistical learning via the alternating direction method of multipliers”. In: *Foundations and Trends® in Machine learning* 3.1 (2011), pp. 1–122.

[13] Nils Breyer, David Gundlegård, and Clas Rydergren. “Cellpath Routing and Route Traffic Flow Estimation Based on Cellular Network Data”. In: *Journal of Urban Technology* (2017), pp. 1–20.

[14] H Bush, Patrick West, and Lynn Michell. “The role of friendship groups in the uptake and maintenance of smoking amongst pre-adolescent and adolescent children: Distribution of Frequencies”. In: *Glasgow, Scotland: Medical Research Council* (1997).

[15] Daniele Casati, Kirill Müller, Pieter J Fourie, Alexander Erath, and Kay W Axhausen. “Synthetic population generation by combining a hierarchical, simulation-based approach with reweighting by generalized raking”. In: *Transportation Research Record: Journal of the Transportation Research Board* 2493 (2015), pp. 107–116.

[16] Duke Network Analysis Center. *AN ERGM TUTORIAL USING R.*

[17] Chen Cheng, Haiqin Yang, Irwin King, and Michael R Lyu. “Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks.” In: *Aaai*. Vol. 12. 2012, pp. 17–23.

[18] Xingyi Cheng, Ruiqing Zhang, Jie Zhou, and Wei Xu. “DeepTransport: Learning Spatial-Temporal Dependency for Traffic Condition Forecasting”. In: *arXiv preprint arXiv:1709.09585* (2017).

[19] David Maxwell Chickering, David Heckerman, and Christopher Meek. “A Bayesian approach to learning Bayesian networks with local structure”. In: *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc. 1997, pp. 80–89.

[20] Eunjoon Cho, Seth A Myers, and Jure Leskovec. “Friendship and mobility: user movement in location-based social networks”. In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2011, pp. 1082–1090.

[21] Nicholas A Christakis and James H Fowler. “Social contagion theory : examining dynamic social networks and human behavior”. In: November 2011 (2013). DOI: 10.1002/sim.5408.

[22] H. Chung, B. P. Flaherty, and J. L. Schafer. “Latent class logistic regression: application to marijuana use and attitudes among high school seniors”. In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 169.4 (2006), pp. 723–743.

[23] Michael J Clay and Robert A Johnston. “Multivariate uncertainty analysis of an integrated land use and transportation model: MEPLAN”. In: *Transportation Research Part D: Transport and Environment* 11.3 (2006), pp. 191–203.

[24] Deborah A Cohen, Brian K Finch, Aimee Bower, and Narayan Sastry. "Collective efficacy and obesity: the potential influence of social factors on health". In: *Social science & medicine* 62.3 (2006), pp. 769–778.

[25] IBM ILOG CPLEX. "V12. 1: User's Manual for CPLEX". In: *International Business Machines Corporation* 46.53 (2009), p. 157.

[26] Tomás De La Barra, Beatriz Pérez, Vera, and N. "TRANUS-J: putting large models into small computers". In: *Environment and Planning B: Planning and Design* 11.1 (1984), pp. 87–101.

[27] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering". In: *Advances in Neural Information Processing Systems*. 2016, pp. 3844–3852.

[28] Dingxiong Deng, Cyrus Shahabi, Ugur Demiryurek, Linhong Zhu, Rose Yu, and Yan Liu. "Latent space model for road networks to predict time-varying traffic". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2016, pp. 1525–1534.

[29] Dingxiong Deng, Cyrus Shahabi, Ugur Demiryurek, and Linhong Zhu. "Situation Aware Multi-Task Learning for Traffic Prediction". In: *Data Mining (ICDM), 2017 IEEE International Conference on*. IEEE. 2017, pp. 81–90.

[30] B. Deylon, M. Lavielle, and E. Moulines. "Convergence of a Stochastic Approximation Version of the EM Algorithm". In: *The Annals of Statistics* 27.1 (1999), pp. 94–128.

[31] Steven Diamond and Stephen Boyd. "CVXPY: A Python-embedded modeling language for convex optimization". In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5.

[32] Alexander Domahidi, Eric Chu, and Stephen Boyd. "ECOS: An SOCP solver for embedded systems". In: *Control Conference (ECC), 2013 European*. IEEE. 2013, pp. 3071–3076.

[33] Donald R Drew. *Traffic flow theory and control*. Tech. rep. 1968.

[34] Elenna Dugundji and Joan Walker. "Discrete choice with social and spatial network interdependencies: an empirical example using mixed generalized extreme value models with field and panel effects". In: *Transportation Research Record: Journal of the Transportation Research Board* 1921 (2005), pp. 70–78.

[35] Elenna Dugundji and Joan Walker. "Discrete Choice with Social and Spatial Network Interdependencies: An Empirical Example Using Mixed Generalized Extreme Value Models with Field and Panel Effects". In: *Transp. Res. Rec.* 1921.1 (2005), pp. 70–78. ISSN: 0361-1981. DOI: 10.3141/1921-09.

[36] Paul Erdős and Alfréd Rényi. "On random graphs". In: *Publicationes Mathematicae Debrecen* 6 (1959), pp. 290–297.

[37] Hugh Everett III. “Generalized Lagrange multiplier method for solving problems of optimum allocation of resources”. In: *Operations research* 11.3 (1963), pp. 399–417.

[38] Bilal Farooq, Michel Bierlaire, Ricardo Hurtubia, and Gunnar Flötteröd. “Simulation based population synthesis”. In: *Transportation Research Part B: Methodological* 58 (2013), pp. 243–263.

[39] Kimon Fountoulakis, David Gleich, and Michael Mahoney. “An optimization approach to locally-biased graph algorithms”. In: *arXiv preprint arXiv:1607.04940* (2016).

[40] Kimon Fountoulakis, Xiang Cheng, Julian Shun, Farbod Roosta-Khorasani, and Michael W Mahoney. “Exploiting Optimization for Local Graph Clustering”. In: *arXiv preprint arXiv:1602.01886* (2016).

[41] Ove Frank and David Strauss. “Markov graphs”. In: *Journal of the american Statistical association* 81.395 (1986), pp. 832–842.

[42] Song Gao, Yu Liu, Yaoli Wang, and Xiujun Ma. “Discovering spatial interaction communities from mobile phone data”. In: *Transactions in GIS* 17.3 (2013), pp. 463–481.

[43] David F Gleich and Michael W Mahoney. “Using local spectral methods to robustify graph-based learning algorithms”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2015, pp. 359–368.

[44] Tom Goldstein, Brendan O’Donoghue, Simon Setzer, and Richard Baraniuk. “Fast alternating direction optimization methods”. In: *SIAM Journal on Imaging Sciences* 7.3 (2014), pp. 1588–1623.

[45] Elizabeth Gordon and Susan L Handy. *Safe and Normal: Social Influences on Formation of Attitudes Toward Bicycling*. Tech. rep. 2012.

[46] William Gropp, Ewing Lusk, Nathan Doss, and Anthony Skjellum. “A high-performance, portable implementation of the MPI message passing interface standard”. In: *Parallel computing* 22.6 (1996), pp. 789–828.

[47] Aditya Grover and Jure Leskovec. “node2vec: Scalable feature learning for networks”. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2016, pp. 855–864.

[48] Avery M. Guest, Jane K. Cover, Ross L. Matsueda, and Charis E. Kubrin. “Neighborhood Context and Neighboring Ties”. In: *City Community* 5.4 (2006), pp. 363–385. ISSN: 1540-6040. DOI: 10.1111/j.1540-6040.2006.00189.x. URL: http://dx.doi.org/10.1111/j.1540-6040.2006.00189.x.

[49] David Gundlegård, Clas Rydergren, Nils Breyer, and Botond Rajna. “Travel demand estimation and network assignment based on cellular network data”. In: *Computer Communications* 95 (2016), pp. 29–42.

[50] David Hallac, Jure Leskovec, and Stephen Boyd. "Network lasso: Clustering and optimization in large graphs". In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM. 2015, pp. 387–396.

[51] Ross A Hammond. "Social influence and obesity". In: *Current Opinion in Endocrinology, Diabetes and Obesity* 17.5 (2010), pp. 467–471.

[52] Ramaswamy Hariharan and Kentaro Toyama. "Project Lachesis: parsing and modeling location histories". In: *International Conference on Geographic Information Science*. Springer. 2004, pp. 106–124.

[53] Kayla De la Haye, Garry Robins, Philip Mohr, and Carlene Wilson. "Obesity-related behaviors in adolescent friendship networks". In: *Social Networks* 32.3 (2010), pp. 161–167.

[54] David Heckerman. "A tutorial on learning with Bayesian networks". In: *Learning in graphical models*. Springer, 1998, pp. 301–354.

[55] David Heckerman, Dan Geiger, and David M Chickering. "Learning Bayesian networks: The combination of knowledge and statistical data". In: *Machine learning* 20.3 (1995), pp. 197–243.

[56] Mikael Henaff, Joan Bruna, and Yann LeCun. "Deep convolutional networks on graph-structured data". In: *arXiv preprint arXiv:1506.05163* (2015).

[57] Carlos Herrera-Yagüe, Christian M Schneider, Thomas Couronne, Zbigniew Smoreda, Rosa M Benito, Pedro J Zufiria, and Marta C González. "The anatomy of urban social networks and its implications in the searchability problem". In: *Scientific reports* 5 (2015).

[58] Cesar A Hidalgo and C Rodriguez-Sickert. "The dynamics of a mobile phone network". In: *Physica A: Statistical Mechanics and its Applications* 387.12 (2008), pp. 3017–3024.

[59] David R Hunter, Mark S Handcock, Carter T Butts, Steven M Goodreau, and Martina Morris. "ergm: A package to fit, simulate and diagnose exponential-family models for networks". In: *Journal of statistical software* 24.3 (2008), nihpa54860.

[60] Kasthuri Jayarajah, Noel Athaide, Vigneshwaran Subbaraju, and Archan Misra. "Detection, Localization and Characterization of Transient, Urban Events using Multi-Modal Information". In: ().

[61] Lucas GS Jeub, Prakash Balachandran, Mason A Porter, Peter J Mucha, and Michael W Mahoney. "Think locally, act locally: Detection of small, medium-sized, and large communities in large networks". In: *Physical Review E* 91.1 (2015), p. 012821.

[62] Shan Jiang, Yingxiang Yang, Siddharth Gupta, Daniele Veneziano, Shounak Athavale, and Marta C González. "The TimeGeo modeling framework for urban motility without travel surveys". In: *Proceedings of the National Academy of Sciences* (2016), p. 201524261.

[63] Jinhee Kim, Soora Rasouli, and Harry Timmermans. “Expanding scope of hybrid choice models allowing for mixture of social influences and latent attitudes: Application to intended purchase of electric cars”. In: *Transportation research part A: policy and practice* 69 (2014), pp. 71–85.

[64] Jinhee Kim, Soora Rasouli, and Harry JP Timmermans. “Investigating heterogeneity in social influence by social distance in car-sharing decisions under uncertainty: A regret-minimizing hybrid choice model framework based on sequential stated adaptation experiments”. In: *Transportation Research Part C: Emerging Technologies* 85 (2017), pp. 47–63.

[65] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907* (2016).

[66] Kevin S Kung, Kael Greco, Stanislav Sobolevsky, and Carlo Ratti. “Exploring universal patterns in human home-work commuting from mobile phone data”. In: *PloS one* 9.6 (2014), e96180.

[67] Renaud Lambiotte, Vincent D Blondel, Cristobald De Kerchove, Etienne Huens, Christophe Prieur, Zbigniew Smoreda, and Paul Van Dooren. “Geographical dispersal of mobile communication networks”. In: *Physica A: Statistical Mechanics and its Applications* 387.21 (2008), pp. 5317–5325.

[68] Leon S Lasdon. *Optimization theory for large systems.* Courier Corporation, 1970.

[69] A. Lawlor, C. Coffey, R. McGrath, and A. Pozdnoukhov. “Stratification structure of urban habitats”. In: *Pervasive Urban Applications at PERVASIVE* (2012).

[70] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. “Graph evolution: Densification and shrinking diameters”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1.1 (2007), p. 2.

[71] Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. “Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters”. In: *Internet Mathematics* 6.1 (2009), pp. 29–123.

[72] R. A. Levine and G. Casella. “Implementations of the Monte Carlo EM Algorithm”. In: *Journal of Computational and Graphical Statistics* 10.3 (2012), pp. 422–439.

[73] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. “Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting”. In: ().

[74] Ziheng Lin, Mogeng Yin, Sidney Feygin, Madeleine Sheehan, Jean-Francois Paiement, and Alexei Pozdnoukhov. “Deep Generative Models of Urban Mobility”. In: (2017).

[75] Bin Liu, Yanjie Fu, Zijun Yao, and Hui Xiong. “Learning geographical preferences for point-of-interest recommendation”. In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM. 2013, pp. 1043–1051.

[76] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. "Recommender systems with social regularization". In: *WSDM '11 Proceedings of the fourth ACM international conference on Web search and data mining* (2011), pp. 287–296.

[77] Jonathan Magnusson and Tor Kvernvik. "Subscriber classification within telecom networks utilizing big data technologies and machine learning". In: *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*. ACM. 2012, pp. 77–84.

[78] Charles F. Manski. "Identification of Social Endogenous Effects: The Reflection Problem". In: *Rev. Econ. Stud.* 60.3 (1993), pp. 531–542. ISSN: 0034-6527. DOI: 10.2307/2298123.

[79] Wesley Mathew, Ruben Raposo, and Bruno Martins. "Predicting future locations with hidden Markov models". In: *Proceedings of the 2012 ACM conference on ubiquitous computing*. ACM. 2012, pp. 911–918.

[80] Daniel McFadden et al. "Conditional logit analysis of qualitative choice behavior". In: (1973).

[81] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[82] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[83] Mark R Montgomery and John B Casterline. "Social learning, social influence, and new models of fertility". In: *Population and Development Review* 22 (1996), pp. 151–175.

[84] Ben-Akiva Moshe, Bierlaire Michel, McFadden Daniel, and Walker Joan. "Discrete Choice Analysis". In: ().

[85] Kevin P Murphy, Yair Weiss, and Michael I Jordan. "Loopy belief propagation for approximate inference: An empirical study". In: *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc. 1999, pp. 467–475.

[86] Ronald C Neath et al. "On convergence properties of the Monte Carlo EM algorithm". In: *Advances in Modern Statistical Theory and Applications: A Festschrift in Honor of Morris L. Eaton*. Institute of Mathematical Statistics, 2013, pp. 43–62.

[87] Mark EJ Newman. "Modularity and community structure in networks". In: *Proceedings of the national academy of sciences* 103.23 (2006), pp. 8577–8582.

[88] Krzysztof Nowicki and Tom A B Snijders. "Estimation and prediction for stochastic blockstructures". In: *Journal of the American Statistical Association* 96.455 (2001), pp. 1077–1087.

[89] J-P Onnela, Jari Saramäki, Jorkki Hyvönen, György Szabó, David Lazer, Kimmo Kaski, János Kertész, and A-L Barabási. "Structure and tie strengths in mobile communication networks". In: *Proceedings of the National Academy of Sciences* 104.18 (2007), pp. 7332–7336.

[90] Jukka-Pekka Onnela, Samuel Arbesman, Marta C González, Albert-László Barabási, and Nicholas A Christakis. "Geographic constraints on social network groups". In: *PLoS one* 6.4 (2011), e16939.

[91] Gergely Palla, Albert-László Barabási, and Tamás Vicsek. "Quantifying social group evolution". In: *Nature* 446.7136 (2007), pp. 664–667.

[92] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2014, pp. 701–710.

[93] Santi Phithakkitnukoon, Zbigniew Smoreda, and Patrick Olivier. "Socio-geography of human mobility: A study using longitudinal mobile phone data". In: *PloS one* 7.6 (2012), e39253.

[94] Susan Pike. "Travel Mode Choice and Social and Spatial Reference Groups: Comparison of Two Formulations". In: *Transportation Research Record: Journal of the Transportation Research Board* 2412 (2014), pp. 75–81.

[95] Pratap S Prasad and Prathima Agrawal. "Movement prediction in wireless networks using mobility traces". In: *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*. IEEE. 2010, pp. 1–5.

[96] Wen Pu, Jaesik Choi, Eyal Amir, and Dorothy L Espelage. "Learning exponential random graph models". In: *Urbana* 51 (2013), p. 61801.

[97] Carlo Ratti, Stanislav Sobolevsky, Francesco Calabrese, Clio Andris, Jonathan Reades, Mauro Martino, Rob Claxton, and Steven H Strogatz. "Redrawing the map of Great Britain from a network of human interactions". In: *PloS one* 5.12 (2010), e14248.

[98] Garry Robins, Philippa Pattison, and Peter Elliott. "Network models for social influence processes". In: *Psychometrika* 66.2 (2001), pp. 161–189.

[99] Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher. "An introduction to exponential random graph (p*) models for social networks". In: *Social networks* 29.2 (2007), pp. 173–191.

[100] Robert W Robinson. "Counting unlabeled acyclic digraphs". In: *Combinatorial mathematics V*. Springer, 1977, pp. 28–43.

[101] K. Roeder, K. G. Lynch, and D. S. Nagin. "Modeling Uncertainty in Latent Class Membership: A Case Study in Criminology". In: *Journal of the American Statistical Association* 94.447 (1999), pp. 766–776.

[102] Feygin S. and Pozdnoukhov A. "Peer Pressure Enables Actuation of Mobility Lifestyles". In: ().

[103] Paul Salvini and Eric J Miller. “ILUTE: An operational prototype of a comprehensive microsimulation model of urban systems”. In: *Networks and Spatial Economics* 5.2 (2005), pp. 217–234.

[104] David Schrank, Bill Eisele, and Tim Lomax. “TTI’s 2012 urban mobility report”. In: *Texas A&M Transportation Institute. The Texas A&M University System* (2012), p. 4.

[105] M Schweinberger and MS Handcock. *Hierarchical exponential-family random graph Models.* Tech. rep. Technical report, Pennsylvania State University [soumis à publication], 2009.

[106] Michael Schweinberger, Mark Handcock, and Pamela Luna. *hergm: Hierarchical Exponential-Family Random Graph Models with Local Dependence.* 2016.

[107] Michael Schweinberger and Mark S Handcock. “Local Dependence in Random Graph Models: Characterization, Properties, and Statistical Inference”. In: *Journal of the Royal Statistical Society, Series B* (2015).

[108] Marco Scutari. “Learning Bayesian networks with the bnlearn R package”. In: *arXiv preprint arXiv:0908.3817* (2009).

[109] Cr Shalizi and Ac Thomas. “Homophily and contagion are generically confounded in observational social network studies”. In: *Sociol. Methods Res.* (2011), pp. 1–27. arXiv: `arXiv:1004.4704v3`. URL: `http://smr.sagepub.com/content/40/2/211.short`.

[110] Li Shi, Guanghua Chi, Xi Liu, and Yu Liu. “Human mobility patterns in different communities: a mobile phone data-based social network approach”. In: *Annals of GIS* 21.1 (2015), pp. 15–26.

[111] Julian Shun and Guy E Blelloch. “Ligra: a lightweight graph processing framework for shared memory”. In: *ACM SIGPLAN Notices.* Vol. 48. 8. ACM. 2013, pp. 135–146.

[112] Julian Shun, Farbod Roosta-Khorasani, Kimon Fountoulakis, and Michael W Mahoney. “Parallel Local Graph Clustering”. In: *arXiv preprint arXiv:1604.07515* (2016).

[113] John Skvoretz and Katherine Faust. “Logit models for affiliation networks”. In: *Sociological Methodology* 29.1 (1999), pp. 253–280.

[114] Tom AB Snijders and Krzysztof Nowicki. “Estimation and prediction for stochastic blockmodels for graphs with latent block structure”. In: *Journal of classification* 14.1 (1997), pp. 75–100.

[115] Lijun Sun and Alexander Erath. “A Bayesian network approach for population synthesis”. In: *Transportation Research Part C: Emerging Technologies* 61 (2015), pp. 49–62.

[116] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. “Line: Large-scale information network embedding”. In: *Proceedings of the 24th International Conference on World Wide Web.* International World Wide Web Conferences Steering Committee. 2015, pp. 1067–1077.

[117] Samuel Thiriot and Jean-Daniel Kant. “Generate country-scale networks of interaction from scattered statistics”. In: *The fifth conference of the European social simulation association, Brescia, Italy.* Vol. 240. 2008.

[118] Paul Waddell. “UrbanSim: Modeling urban development for land use, transportation, and environmental planning”. In: *Journal of the American Planning Association* 68.3 (2002), pp. 297–314.

[119] Martin J Wainwright, Michael I Jordan, et al. “Graphical models, exponential families, and variational inference”. In: *Foundations and Trends® in Machine Learning* 1.1–2 (2008), pp. 1–305.

[120] J. L. Walker and J. Li. “Latent lifestyle preferences and household location decisions”. In: *Journal of Geographical Systems* 9.1 (2007), pp. 77–101.

[121] Joan Leslie Walker. “Extended discrete choice models: integrated framework, flexible error structures, and latent variables”. PhD thesis. Massachusetts Institute of Technology, 2001.

[122] Fergal Walsh and Alexei Pozdnoukhov. “Spatial structure and dynamics of urban communities”. In: (2011).

[123] G. C. .G. Wei and M. A. Tanner. “A Monte Carlo Implementation of the EM Algorithm and the Poor Man’s Data Augmentation Algorithms”. In: *Journal of the American Statistical Association* 85.411 (1990), pp. 699–704.

[124] Peter Widhalm, Yingxiang Yang, Michael Ulm, Shounak Athavale, and Marta C González. “Discovering urban activity patterns in cell phone data”. In: *Transportation* 42.4 (2015), pp. 597–623.

[125] Derry Wijaya, Partha Pratim Talukdar, and Tom Mitchell. “Pidgin: ontology alignment using web text as interlingua”. In: *Proceedings of the 22nd ACM international conference on Information & Knowledge Management.* ACM. 2013, pp. 589–598.

[126] Alan S Willsky, Erik B Sudderth, and Martin J Wainwright. “Loop series and Bethe variational bounds in attractive graphical models”. In: *Advances in neural information processing systems.* 2008, pp. 1425–1432.

[127] Ling Heng Wong, Philippa Pattison, and Garry Robins. “A spatial model for social networks”. In: *Physica A: Statistical Mechanics and its Applications* 360.1 (2006), pp. 99–120.

[128] Cathy Wu, Jérôme Thai, Steve Yadlowsky, Alexei Pozdnoukhov, and Alexandre Bayen. “Cellpath: Fusion of cellular and traffic sensor data for route flow estimation via convex optimization”. In: *Transportation Research Part C: Emerging Technologies* 59 (2015), pp. 111–128.

[129] Xiaolin Yang, Stephen E. Fienberg, Alessandro Rinaldo, Han Liu, and Michael Rosenblum. *blkergm: Fitting block ERGM given the block structure on social networks.* R package version 1.1. 2014. URL: `https://CRAN.R-project.org/package=blkergm`.

[130] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. "Revisiting semi-supervised learning with graph embeddings". In: *arXiv preprint arXiv:1603.08861* (2016).

[131] Jihang Ye, Zhe Zhu, and Hong Cheng. "What's your next move: User activity prediction in location-based social networks". In: *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM. 2013, pp. 171–179.

[132] Xin Ye, Karthik Konduri, Ram M Pendyala, Bhargava Sana, and Paul Waddell. "A methodology to match distributions of both household and person attributes in the generation of synthetic populations". In: *88th Annual Meeting of the Transportation Research Board, Washington, DC*. 2009.

[133] Mogeng Yin, Madeleine Sheehan, Sidney Feygin, Jean-Francois Paiement, and Alexei Pozdnoukhov. "A generative model of urban activities from cellular data". In: *IEEE Transactions on Intelligent Transportation Systems* (2017).

[134] Mogeng Yin, Madeleine Sheehan, Sidney Feygin, Jean-François Paiement, and Alexei Pozdnoukhov. "A generative model of urban activities from cellular Data". In: *IEEE Transactions on Intelligent Transportation Systems* (2017).

[135] Jing Yuan, Yu Zheng, and Xing Xie. "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. 2012, pp. 186–194.

[136] Nicholas Jing Yuan, Yu Zheng, and Xing Xie. "Segmentation of urban areas using road networks". In: *MSR-TR-2012–65, Tech. Rep.* (2012).

[137] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. "Time-aware point-of-interest recommendation". In: *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM. 2013, pp. 363–372.

[138] Danqing Zhang, Kimon Fountoulakis, Junyu Cao, Mogeng Yin, Michael Mahoney, and Alexei Pozdnoukhov. "Social Discrete Choice Models". In: *arXiv preprint arXiv:1703.07520* (2017).

[139] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. "Urban computing: concepts, methodologies, and applications". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 5.3 (2014), p. 38.

[140] Yu Zheng, Yanchi Liu, Jing Yuan, and Xing Xie. "Urban computing with taxicabs". In: *Proceedings of the 13th international conference on Ubiquitous computing*. ACM. 2011, pp. 89–98.

[141] Denny Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. "Learning with local and global consistency". In: *Advances in neural information processing systems*. 2004, pp. 321–328.

[142] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. “Semi-supervised learning using gaussian fields and harmonic functions”. In: *Proceedings of the 20th International conference on Machine learning (ICML-03)*. 2003, pp. 912–919.

[143] Yi Zhu and Joseph Ferreira. “Synthetic population generation at disaggregated spatial scales for land use and transportation microsimulation”. In: *Transportation Research Record: Journal of the Transportation Research Board* 2429 (2014), pp. 168–177.