5-2018

Learning latent characteristics of locations using location-based social networking data

Thanh Nam DOAN
Singapore Management University, tndoan.2012@phdis.smu.edu.sg

LEARNING LATENT CHARACTERISTICS OF LOCATIONS USING LOCATION-BASED SOCIAL NETWORKING DATA

THANH-NAM DOAN

SINGAPORE MANAGEMENT UNIVERSITY
2018

Learning latent characteristics of locations using Location-based Social Networking Data

by
Thanh-Nam Doan

Submitted to School of Information Systems in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Information Systems

Dissertation Committee:

Ee-Peng Lim (Supervisor/Chair)
Professor of Information Systems
Singapore Management University

Baihua Zheng
Associate Professor of Information Systems
Singapore Management University

Hady W. Lauw
Assistant Professor of Information Systems
Singapore Management University

Xiaoli Li
Department Head (Data Analytics)
Institute for Infocomm Research, A*STAR

Singapore Management University
2018

Learning latent characteristics of locations
using Location-based Social Networking Data

by

Thanh-Nam Doan

Abstract

This dissertation addresses the modeling of latent characteristics of locations to describe the mobility of users of location-based social networking platforms. With many users signing up location-based social networking platforms to share their daily activities, these platforms become a gold mine for researchers to study human visitation behavior and location characteristics. Modeling such visitation behavior and location characteristics can benefit many useful applications such as urban planning and location-aware recommender systems. In this dissertation, we focus on modeling two latent characteristics of locations, namely *area attraction* and *neighborhood competition* effects using location-based social network data. Our literature survey reveals that previous researchers did not pay enough attention to area attraction and neighborhood competition effects. Area attraction refers to the ability of an area with multiple venues to collectively attract check-ins from users, while neighborhood competition represents the need for a venue to compete with its neighbors in the same area for getting check-ins from users.

In this dissertation, we firstly gather the location-based social networking data generated by Foursquare users from two big cities in Southeast Asia: Singapore and Jakarta. To generalize our findings, we also employ the Gowalla data of users from New York City. We then embark on a data science study of area attraction, neighborhood competition, and other user and location related effects including spatial homophily, social homophily, distance effects. Since the interaction between users and locations is a complex process involving mul-

tiple effects, we propose several novel models that incorporate latent location and social factors in the generation of users' visitation. These models utilize a range of different techniques, including PageRank, Bayesian reasoning, matrix factorization, and neural networks. Each model is evaluated through extensive experiments and the results show that neighborhood competition and area attraction effects contribute to more accurate modeling and prediction of users' visitation to locations.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I am grateful to my advisor, Professor Ee-Peng Lim, for his continuous support, guidance, and encouragement, and for providing me with the best working environment throughout my PhD study.

I appreciate useful comments and feedbacks from members of my dissertation committee: Professor Ee-Peng Lim, Associate Professor Baihua Zheng, Assistant Professor Hady W. Lauw, and Dr. Xiaoli Li.

I was fortunate to have supports from my fellow researchers and friends in Living Analytics Research Centre (LARC): Freddy Chong Tat Chua, Philips K. Prasetyo, Agus T. Kwee, Ibrahim Nelman Lubis, Arinto Murdopo, Tuan-Anh Hoang, and Luu Minh Duc.

I am indebted to administrative staff in School of Information Systems: Ong Chew Hong and Seow Pei Huan; and the staff in LARC: Chua Kian Peng, Cheong Su Fen, Angela Kwek Renfeng, Sheryl Chen, Phoebe Yeo, Alenzia Wong Poh Luan, Fong Soon Keat, Desmond Yap, Janice Tan, Jamie Chia Yih Min and Nancy Beatty.

I would like to thank Singapore Management University for the scholarships and financial supports for my study. My research work is funded by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative.

Publications

Publications based on the dissertation. Listed by reverse chronological order:

1. Thanh-Nam DOAN and Ee-Peng LIM, *PACELA: A Neural Framework for User Visitation in Location-based Social Networks*, User Modeling, Adaptation and Personalization, UMAP 2018. (Chapter 8)

2. Thanh-Nam DOAN and Ee-Peng LIM, *Modeling Location-based Social Network Data with Area Attraction and Neighborhood Competition*, (under review in Data Mining and Knowledge Discovery, DMKD 2018). (Chapter 6)

3. Thanh-Nam DOAN and Ee-Peng LIM, *Modeling Check-In Behavior with Geographical Neighborhood Influence of Venues*, International Conference on Advanced Data Mining and Applications, ADMA 2017. (Chapter 7)

4. Thanh-Nam DOAN and Ee-Peng LIM, *Attractiveness versus Competition: Towards an Unified Model for User Visitation*, International on Conference on Information and Knowledge Management, CIKM 2016. (Chapter 5)

5. Thanh-Nam DOAN, Freddy Chong Tat CHUA, and Ee-Peng LIM, *Mining Business Competitiveness from User Visitation Data*, International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction, SBP 2015. (Chapter 4)

Other publications during PhD study

Thanh-Nam DOAN, Freddy Chong Tat CHUA, and Ee-Peng LIM, *On neighborhood effects in location-based social networks*, International Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT 2015.

Dedicated to my family

Chapter 1

Introduction

1.1 Motivation

The popularity of smartphones and wearable devices in recent years has propelled the growth of location based social networking (LBSN) sites where users publish and share their visits (or check-ins) to different venues. For example, Foursquare is used by 50 millions users each month and it covers more than 65 million venues around the world. These users have generated 8 billion check-ins worldwide [1]. The check-in feature does not exist on LBSN sites only. Many major online social networking sites also adopt this feature to enrich their social interaction. For instance, Facebook Places is a new feature which allows Facebook users to share visited locations in their timelines. Similarly, Twitter's users can associate their tweets with geo-locations.

With so much of data generated by LBSN and social media users, they provide unprecedented opportunities for researchers to study the visitation patterns of users and interaction between users and locations. These data also capture various effects of user visitation behavior which can be attributed to several latent location and user factors. It is this kind of datasets covering granular level user and venue activity data that allow new models to be developed and evaluated.

[1] https://foursquare.com/about - April 2017

The data observed at LBSN sites record several types of user behaviors as shown in Table 1.1. The table shows that users are offered a wide range of actions in LBSNs. Users can perform online *check-ins* on locations in LBSNs when they visit them. The behavior of users making friends or following one another is called *social networking*. Users can express their opinions by writing reviews on venues or rating them. These are known as the *reviewing* and *rating* behaviors respectively. Finally, LBSN sites support *media sharing* behaviors as users upload photos or videos and share with friends. Some LBSN sites like Foursquare create games when users visit locations multiple times (e.g. *gaming* behavior). Since the activity of users in LBSNs is multi-modal, LBSN datasets are a great resource for researchers to study behavior of users particularly their interaction with locations.

Table 1.1: Common behaviors of users in LBSNs.

Behavior type	**Description**
Checking in	Users' declaration of visits to locations
Social Networking	Users' connection with other users by following and befriending them
Reviewing	Users writing reviews on venues
Rating	Users' rating on venues
Media Sharing	Uploading of photos/videos
Gaming	Earning of badges/awards

Analyzing LBSN information not only gives us insights of user behavioral patterns, but also reveals interesting characteristics of locations which benefit several applications [94, 77, 18, 79, 81, 37, 16] such as urban planning, location recommendation and customer relationship management. Urban planners could identify popular areas or locations before building new roads or new subway stations. Smartphone apps could recommend some restaurants or venues around a user's location by aggregating reviews from LBSNs. Business venue owners can benefit by receiving feedback from their customers via LBSNs. With LBSNs, the owners could monitor customer feedback in real time and even engage them proactively.

Although there are many research works [14, 4, 11] studying check-in behaviors of users in LBSNs, they focus on user behavior under the effects of *spatial homophily* [35, 60], *social homophily* [51, 52] and *distance effect* [54, 14, 84]. *Spatial homophily* suggests that users(or venues) that are nearby one another are likely to be similar in their check-in venues(or users); while *social homophily* states that users and their friends share more common venue preference than between the users and the other strangers. The general idea of *distance effect* is that users are likely to visit venues nearby their home locations rather than venues farther away from their home locations. However, there are other important effects which are not widely studied but have significant influence on users' visitation. *Neighborhood competition* effect is one of these effects. Neighborhood competition assumes that each venue has to compete with its neighbors in the same area for check-ins. In other words, neighborhood competition of venues could be viewed as a race among nearby venues to attract visitors. It is an important effect because user time and attention are limited but inside one neighborhood, there are a lot of venues for users to perform visitation. For this reason, users usually focus on the best venue to visit so inside one cluster, venues must compete with surrounding others to gain the attention of users. For example, shopping malls nearby each other are expected to fight for their shares of customer visits.

Another effect which is not well studied in previous works is *area attraction*. This effect is based on the principle of *"The whole is greater than the sum of its parts"*. Area attraction suggests that the total number of visits to an area is larger than the sum of visits each venue in the area can individually attract. In other words, users visit a venue in an area not only because of this venue alone but also the surrounding area that includes other neighboring venues. For example, the McDonald branches in downtown area attract more users' visitation than the ones in farther away areas despite their similar quality. This suggests that downtown area is more attractive to users than the farther

away ones. Previous works such as Karamshuk *et. al.* [40] measured the attractiveness of an area by the number of visitation of users. However, using check-in popularity is not an accurate method to capture area attraction effect. For instance, one area has large number of check-ins but most of check-ins is concentrated into one venue, and a few check-ins on other venues. We cannot conclude that the area is attractive despite of its huge volume of visitations. Modeling *area attraction* as well as measuring *area attractiveness* could help business owners to understand the attraction of their business areas. Therefore, these owners could develop suitable strategies to increase visits to their business venues. For instance, the business owners in the same area may together advertise to lure more customers. Another application is to aid urban planners to decide which areas to redevelop so as to improve the spread of commercial opportunities across a city.

1.2 Research Objectives

In this dissertation, we therefore aim to (i) use data science study to illustrate the different effects (with special focus on *neighborhood competition* and *area attraction*) of user behavior as they perform check-ins in LBSNs; (ii) identify and learn the latent factors relevant to these effects as we model the user check-in behavior, and (iii) apply our proposed models to real datasets so as to evaluate their performance. The first objective is covered by our *empirical research* while the latter two objectives are achieved by research on *modeling latent user and venue attributes* that are related to these effects. In the following, we cover the two topics in Sections 1.2.1 and 1.2.2.

1.2.1 Empirical Research

There are many empirical studies [51, 52, 14, 54] on LBSN data to study the different behavioral effects on check-in activities. To conduct empirical

research, we crawl the check-in data of specific cities since our research study requires all public detailed data of users living in a city. However, to ensure the robustness of our findings, we also use another dataset available to researchers to verify our empirical findings.

Since the dissertation focuses on the two effects namely *neighborhood competition* and *area attraction*, we would like to design empirical analyses to illustrate these effects using LBSN datasets. There are several issues to address prior the analyses. First of all, the two effects do not receive enough attention from researchers, and there are no well established studies on them. There is a lack of formal definitions and theories about them. Some previous works [40, 59] used number of check-ins of locations and areas to represent *neighborhood competition* and *area attraction* but as shown before, these two effects cannot be modeled by popularity. As one of the first works, we need to determine appropriate measures for showing the existence of these effects. Secondly, check-ins made by users in LBSNs is a mixture of multiple effects and the exact interaction among effects is still the open question. Hence, we need to isolate *neighborhood competition* and *area attraction* from other effects in our empirical analyses.

We also want to verify the earlier findings on other effects such as spatial homophily or distance effect which involve home locations of users. Previously, most works did not consider the use of actual user home location even when it is an important component of these effects. Due to privacy concerns, many users do not want to reveal their exact home locations. In the absence of actual user home location, some previous works [54, 4] used high-level locations (e.g. city level) as home locations of users and studied the effects at the coarse granularity level. Others [51, 69, 14] resorted to estimating the home locations of users. Specifically, Li *et. al.* [51] estimated home locations of users by using recursive grid search method [13]. Qu *et. al.* [69] and Cho *et. al.* [14] approximated home locations of users and then used them to study spatial

homophily and distance effects as well as other features such as neighbors of users. Coarse-grained home locations of users are not appropriate for our empirical analysis of users within a city performing check-ins at the fine-grained venue level. For this reason, we focus on analysis using the true home locations of users. This marks the main difference between our works and other previous ones analyzing spatial homophily and distance effects.

1.2.2 Modeling Latent Properties of Locations

We want to model the latent attributes relevant to *neighborhood competition* and *area attraction*. We also aim to combine these latent attributes with those relevant to *distance effect* and *social/spatial homophily* in the modeling of check-in behavior of users. The first obstacle of this modeling research is a lack of formal definitions of these effects. Thus, we need to define and formalize them clearly. Secondly, from the effects, we want to derive relevant latent factors, determine their inter-relationships and incorporate them into new models of check-in behavior. We need models that incorporate *neighborhood competition* and *area attraction* effects as well as the more studied effects such as distance effect. Furthermore, depending on the modeling approach, i.e. Bayesian reasoning [54, 85], matrix factorization [27, 52, 53], different model products can be developed. Finally, the research would not be complete without evaluation. With an absence of ground truth data, we have to consider task based evaluation, which involves a number of prediction tasks including check-in prediction, home locations prediction and venue ranking. Moreover, the evaluation has to cover the performance of the models under different configuration settings.

Table 1.2 summarizes our proposed models and the set of effects considered by these models. First of all, we want to model *neighborhood competition* without using latent factors. We develop PageRank-based and Bayesian models that incorporate the effect of *neighborhood competition*. Our proposed

Table 1.2: The summary of technique and set of effects in each model.

Type of Model	**Chapter**	**Model**	**Effects**
Models without latent factors	4	PageRank	Neighborhood Competition
	5	Bayesian Reasoning	- Neighborhood Competition - Area Attraction - Distance Effect
Models with latent factors	6	Bayesian Reasoning + Matrix Factorization	- Neighborhood Competition - Area Attraction - Social Homophily
	7	Matrix Factorization	- Neighborhood Competition - Spatial Homophily - Social Homophily
	8	Neural Network	- Neighborhood Competition - Area Attraction - Social Homophily

Bayesian model is flexible enough to also include *area attraction* and *distance effect*. Secondly, we study *neighborhood competition* considering latent factor. Our first proposed latent factor model adopts matrix factorization to factorize visitation of users to venues into user and venue latent factors. Our second proposed latent factor model also adopts user and venue latent factors but it considers the extrinsic factors of venues to enhance the model expressiveness. The last model combines user and venue latent factors with user and venue embedding vectors under the neural network framework to further improve the prediction performance.

1.3 Contribution

Our contribution in this proposal could be summarized as follows:

Empirical Research:

- We have collected the Foursquare data in large scale via Twitter API. Moreover, from the crawled data, we propose a method to identify the exact home location of a subset of users. The home locations are venues users tagged as homes and this venue level home location distinguishes our work from other previous works which infer the approximate home locations of users [14, 51, 69] or assume city level home locations [54, 4].

- In our research on neighborhood competition, we conduct experiment to illustrate the existence of neighborhood competition among venues. This is one of the first studies on neighborhood competition. We propose grouping venues into areas as an appropriate way to measure the neighborhood competitiveness of venues. To construct areas, we divide an entire city into non-overlapping grid areas.

- Area attraction is another effect we formally define and study. In order to reveal the attractiveness of areas, we examine the branches of fast food chain within dense areas and sparse ones. Areas are constructed by the same way as in the study of neighborhood competition and we then measure the distance between users and fast food branches within areas. From the result, we observe that dense areas attract users farther away than sparse ones and it is a clear signal of area attraction. Therefore, the finding clarifies the impact of area attraction effect to users' visitation in LBSNs.

- Using the exact home locations of users, we revisit other effects such as social/spatial homophily and distance effect which have been conducted in previous works. Therefore, we could verify the earlier findings at a different granularity level.

To ensure the generalization of our analysis results, we apply the analysis to public dataset to verify our finding. Since other analyses require home location, the public dataset is employed on the research of social homophily and neighborhood competition effects only.

Modeling Latent Properties of Locations:

We propose several models that utilize a wide range of techniques including PageRank, Bayesian reasoning, matrix factorization, and neural network to model the combination of effects in LBSNs. Since there are multiple effects affecting check-in behavior of users, each model can handle a subset of effects.

- We start with PageRank to model *neighborhood competition* effect. PageRank was originally designed to compute the importance of websites based on directed links inside the pages [46]. To keep our model simple, only *neighborhood competition* is considered in this case. We formalize the competition of venues and their neighbors as a transition graph. We then define a special PageRank model to score venues by their global competitiveness. Our evaluation results empirically show that this model produces competitiveness measures different from popularity measures.

- Next, we propose a Bayesian reasoning model called *VAN* to capture the impact of a few effects including neighborhood competition, distance effect and area attraction effects. It is one of the first work which models the impact of neighborhood competition and area attraction on check-in behavior. We show that *VAN* model is able to learn the home location of users. *VAN* model also derives competitiveness of venues in LBSNs. Last but not least, *VAN* model outperforms several baseline models in check-in prediction task and home location prediction task.

- To avoid the home location assumption and to consider user preference in venues, we develop a new model that improves over VAN by not requiring user home locations to be known and by modeling user latent preference using a matrix factorization modeling approach. Specifically, each user or venue is represented by a latent feature vector and the check-in of a user to a venue depends on three effects: *preference matching of a user and a venue*, *neighborhood competition* and *area attraction*.

- In the next work, we model the effects of *neighborhood competition*, *spatial homophily* and *social homophily*. We propose a new matrix factorization based model named Extended Neighborhood Matrix Factorization (EN_MF). Besides the vector of intrinsic vector, each venue has the

vector of extrinsic characteristic to model the competition with its neighborhood to gain the attention of users. From our extensive experiments, we observe that our model actually improves the performance of check-in prediction task over baselines. Moreover, we also draw the conclusion that *neighborhood competition* effect contributes more to the accuracy of check-in prediction task than *spatial homophily*.

- To leverage on the predictive power of deep learning model approach [48, 30], we propose Preference And Context Embedding with Latent Attributes (PACELA) which is a neural framework for modeling check-in behavior. Particularly, PACELA learns the embeddings space for the user and venue data as well as the latent attributes of both users and venues. We use a probabilistic matrix factorization-based technique to infer user and venue latent attributes, considering the user visitation decisions under the effect of *area attraction*, *neighborhood competition*, and *social homophily*. PACELA also includes a deep learning neural network to combine both embedding and latent features to predict if a user performs check-in on a location. Our experiments on three different real world datasets show that PACELA yields the best check-in prediction accuracy against several baseline methods.

1.4 Dissertation Structure

The rest of this dissertation is divided as follows. Chapter 2 surveys previous works which are related to my research. Chapter 3 introduces the datasets and also their properties inside. Chapter 4 introduces a model to formalize the competitiveness of venues by modifying PageRank model. Chapter 5 presents the VAN model which captures the neighborhood competition, area attractiveness and distance effect. The model is evaluated under three applications: home location prediction, venues ranking and check-in prediction tasks. Chapter 6

employs matrix factorization based model to improve the VAN model described in Chapter 5. Chapter 7 presents another matrix factorization framework to incorporate neighborhood competition, spatial homophily as well as social homophily. The next chapter 8 describes PACELA a neural framework to understand the check-in behavior of users in LBSNs. Lastly, Chapter 9 provides the conclusion of the dissertation and the future direction research.

Chapter 2

Related Works

In this chapter, we survey the works which study user check-in behavior under a variety of behavioral effects as well as the associated models.

2.1 Modeling Latent Topics of Users and Venues

Before the era of LBSNs, GPS data of human movement have been used by researchers to study movement behaviors [96, 44, 61, 76, 45, 56, 92, 29]. GPS data however do not reveal the venues users have visited. By capturing venue information, LBSNs allow researchers to investigate venue properties and the interaction between venues and users [58, 68, 89] thus leading to the development of new models for check-in behavior. The visitation of users to venues is an outcome of multiple effects to be introduced in Section 2.2. Three of the simplest effects are *user preference*, *venue preference* and *activity content* which only depend on the nature of users and venues respectively.

User topical preference: The topic preference of users in LBSNs refers to the different tastes users have which guide them to visit some specific types of locations. Scellato *et. al.* [74] assume that two users who make check-ins into the same venues share common taste or preference. The likelihood of

them contacting each other in the future is therefore higher than that between two random users. The authors studied the above observation by extracting some features such as the number and the fraction of common places between two users. Bao *et. al.* [5] assumed that each user has different preference for different types of locations. For example, food lovers are likely to focus on restaurants while tourists will pay attention to sightseeing. Therefore, the authors used data from LBSNs to infer the *weight* of each user to each type of venues. In other words, the large value weight represents the high preference of a particular user to a venue. Ye *et. al.* [88] predicted the next locations of users by dividing the selection of users into two steps: (i) users select the category of their next locations, (ii) then, they visit the locations based on the estimated category distribution. The authors used *Hidden Markov Model* (HMM) [2] to map the preferences of each user to categories and then his/her venue choice to category.

Venue topics or types: The visitation of users to venues in LBSNs is driven by not only *user preference* but also *venue type*. Different types of venues attract different types of users. For this reason, Cranshaw *et. al.* [19] used entropy to model the diversity of locations, and they further linked the property to the social interaction at venues. Some previous works such as [12, 52, 51] considered venue preference as latent features so they apply matrix factorization to user-venue check-in matrix to infer the venue preference. Hu *el. al.* [34] adopted *Latent Dirichlet Allocation (LDA)* [8] to understand the venue preference. Specifically, the authors considered each venue as a *document* and the tags associated to a venue as its *words*. Then, they applied *LDA* to understand the topic distribution of each venue. Li *et. al.* [55] conducted the large scale analysis of venues in Foursquare to get the insight of popularity and venues' properties. One interesting finding from their work is that a venue is likely to attract users if it has enough information (e.g. name, category) available on LBSNs.

Content-driven methods: As mentioned in Chapter 1, users can generate activity contents such as tips, ratings and photos for venues in LBSNs. Although not all venues have contents from users, they still contribute an important dimension to study venues since they express users' opinion. For example, a user could check-in to a venue but it does not mean he/she likes this venue since his/her review is bad [15]. Yang *et. al.* [87] and Gao *et. al.* [27] included the sentiment analysis to matrix factorization technique to strengthen the performance of their model to predict check-ins between users and venues. Hu *et. al.* [34] modified *Latent Dirichlet Allocation (LDA)* model [8] to handle content of users for point-of-interest recommendation. Pontes *et. al.* [67] explored *tips*, *dones*, and *mayorships* of Foursquare users. These features are offered by Foursquare as the rewards for users if they share their visits frequently enough. According to this paper, the activity of users is at the same city-level with their home locations. Other papers [32, 1, 31, 39, 93, 17] proposed probabilistic graphical models to incorporate the content of locations with their regions. Their purposes are to model the human check-in behavior as well as predicting users' locations.

2.2 Taxonomy of Effects in Location-based Social Networks

There are three effects of users' visitation which are widely studied in LBSN: *spatial homophily*, *social homophily* and *distance effect*. Moreover, we also mention *neighborhood competition* and *area attraction* effects which are rarely used in understanding users' visitation. Table 2.1 classifies previous works by the combinations of effects they consider in the model development. Particularly, each cell of Table 2.1 contains works which combine the effects in the vertical and horizontal axis. Some works only use one effect so they are in cells whose vertical axis is similar to horizontal axis. To the best of our knowledge,

Table 2.1: Taxonomy of effects in Location based social networks.

	Spatial Homophily		Social	Distance Effect		Neighborhood
	Venue Aspect	User Aspect	Homophily	User Preference	Venue Influence	Competition
Hu *et. al.* [35]	✓					
Liu *et. al.* [60]	✓				✓	
Le Falher *et. al.* [47]	✓					
Gao *et. al.* [28]		✓	✓			
Li *et. al.* [51]		✓	✓			
Backstrom *et. al.* [4]		✓	✓			
Li *et. al.* [52]	✓	✓	✓			
Cheng *et. al.* [12]			✓			
Cho *et. al.* [14]			✓	✓		
Noulas *et. al.* [63]			✓	✓		
Ye *et. al.* [91]			✓		✓	
Chang *et. al.* [11]		✓	✓	✓		
Ye *et. al.* [90]		✓	✓	✓		
Qu *et. al.* [69]				✓		
Tasse *et. al.* [78]				✓		
Li *et. al.* [54]		✓	✓		✓	
Huff [36]					✓	
Liu *et. al.* [59]						✓

there are no previous works related to *area attraction* effects so Table 2.1 does not contain any works for this effect.

2.2.1 Distance Effect

The traveling distances of users are limited and hence users tend to visit nearby venues rather than farther ones. There are two factors inside this effect: *User Preference* and *Venue Influence*. Formally, *User Preference* represents the preferred distance between users and their check-in venues as well as their preferred venue types. The latter, *Venue Influence*, models the selection of users under the consequence of distance between users and venues and influence of venues.

Chang *et. al.* [11] plotted the distribution of check-ins corresponding to several factors like gender of users, temporal information of check-ins using Facebook Places data. They derived multiple features from profile of users (gender of users) or their friends (number of check-ins of friends) or place latent topics using *latent Dirichlet allocation(LDA)* model [8]. The authors then evaluated their proposed method based on linear regression to predict the check-ins of users to venues. According to their result, distance between

users and venues contributes significantly to the prediction.

Ye *et. al.* [90] showed that 87.7% of friends in their LBSN data share nothing in common and concluded not all social connection contributes to users' check-in behavior. They also showed that if a user and his friends live nearby, they are likely to share more commonly visited venues. Based on these observations, they used linear regression to predict the check-ins between users and venues based on the assumption that nearby friends will affect the venue choices of users rather than faraway friends. Moreover, the power law distribution is used to model the probability of users' making check-ins to venues according to the distance between them. Their experiment showed that using spatio-social homophily could lead to accurate check-in prediction between users and venues in LBSNs.

To study the distance effect, there are research works that recover users' home locations from their check-ins. Tasse *et. al.* [78] performed clustering [25] and recursive grid search [13] on a user's tweets generated during the nights to predict the home locations of users. To evaluate their home location prediction methods, they conducted a small scale user study to obtain user locations and their Foursquare check-ins.

In the work by Li *et. al.* [54] which modeled the influence of venues on user check-ins, distance effect has been used to derive the degree of influence of a venue has on users living at different distances away. Specifically, each venue is associated with a Gaussian distribution whose mean is the venue's location and the variance represents its influence. The higher the variance, the more attractive the venue is to the users but this attractiveness decreases with increasing distance between users and the venue. For modeling the social/spatial homophily, they used the same assumption with Backstrom *et. al.* [4] (i.e. users live near to their friends) but they generalized for directional relationship in social networks. However, they also included the new assumption that users in social network mention venues near to their home location.

The more a user mentions a venue, the more likely this venue is close to the home location of the user. Consequently, they associate each user in LBSN to a Gaussian distribution whose mean is user's location and variance is the influence scope of the user. The higher the variance, the higher influence of user is. From that model, they could infer the home location of users at city level instead of precise location. The other applications of their model are that they could rank venues and users based on the influence to other users inside LBSNs.

Huff [36] used distance effect when he modeled the attractiveness of venues. In his model, both a venue area size and travel distance made by its visitors are the two main variables to derive the venue attraction. He assumed that the size of a shopping mall represents its influence on users' selection. His work has some limitations: the size of shopping malls and distance between users and shopping malls are not available in LBSN data. Qu *et. al.* [69] generalizes the work of Huff [36] by using multiple clusters to model the movement of users. Firstly, they replaced time driving distance by the actual distance between users and venues. Moreover, it is the first work which applied Trade Area Analysis (TAA) [36] to location data of users. Furthermore, they also measured the users' preference missing from the Huff's model.

2.2.2 Social Homophily

Similar to other social networks, LBSNs allow users to have social connection with others. In the context of LBSNs, social homophily refers to users who are socially connected and are expected to visit similar venues.

Besides modeling impact of the users' friends to the check-in of users, Li *et. al.* [52] considers the check-ins of two hop away friends to users' check-ins. The authors assumed that check-in between a user and a venue is influenced under two factors (i) distance between a user and a venue, and (ii) the influence of his/her friends and his/her friends of friends (two hop away friends). The

former is modeled by power law distribution while the latter contains two assumptions (i) distance between two users, and (ii) the impact of friends and two hop away friends. The authors used power law distribution for the first assumption and matrix factorization for the second one. Specifically, they construct user-user matrix whose each cell T_{ij} indicates the preference propagated from user j to user i. The observed value of T_{ij} represents the frequency that user i repeats the check-ins of his friends j. T_{ij} is computed by the convex combination of direct influence of friend j to user i and influence of friends of user j to user i.

Cheng *et. al.* [12] modeled social homophily using matrix factorization. Their analysis showed that less than 10% of a user's check-ins is visited by his/her friends. Thus, the social relationship does not contribute much to users' visitation but this effect should not be excluded from the modeling users' check-ins. Since a user and his/her friends share some common preference so the latent vectors of a user and the ones of his/her friends should be similar. Therefore, social homophily is incorporated as a regularization term of matrix factorization technique. As users spend most of their time around multiple activity centers such as work and home, the authors developed multiple matrix factorization models combined with multiple regularization terms to capture social homophily.

Cho *et. al.* [14] used social homophily as well as the periodic movement of users to predict check-ins between users and venues. They proposed a model which captures the two-state behavior of users. First of all, they inferred the home location of each user using grid search [73]. They assumed the check-ins follow power law distribution over the distance from his home to the visited locations. The authors also illustrated the relation between users' mobility and their friendship. The final part of their empirical work showed the temporal and geographic periodicity of users' movement behavior. From the observation that users perform a check-in in the home or work cluster depending on time of

the day, they proposed *Periodic Mobility Model(PMM)* and its variant *Periodic & Social Mobility Model(PSMM)* which considers social network information. Their models could predict the exact home and work locations of users but the home is selected based on the time of check-ins inside the *home* cluster.

Similar to Cho *et. al.* [14], Noulas *et. al.* [63] also took advantage of social homophily and distance effect from users to venues to model check-in behavior of users in LBSNs. They evaluate their model by predicting the *next check-ins* of users. Their methods are based on linear ridge regression [6] and M5 decision tree [70].

Ye *et. al.* [91] used data from Foursquare and Whrrl to study the visitation of users to venues under the impact of social homophily, distance effect and user preference. The authors argued that users and their friends have similar behavior that leads to correlated check-in behaviors. To model the impact of social homophily, they proposed two methods (i) social influence weight, and (ii) random walk with restart (*RWR*) [80]. The social influence weight of a user i and one of his friends v is formalized by the convex combination of (1) Jaccard similarity score between friend set of i and v and (2) Jaccard similarity score of check-in venues between i and v. In *RWR*, one constructs a graph where each node is a user and each edge between node i and node v is weighted by the similarity interaction of the two users. The stationary probability of node i given a starting node k denotes the social influence weight of user k to user i. The authors modeled distance effect as a power law distribution of distance from users to check-in venues. They then combined all features together to derive the probability of user i performing a check-in to venue j.

2.2.3 Spatial Homophily

Spatial homophily exists in two aspects: *venue aspect* and *user aspect*. In the user aspect, users are likely to be similar to others living nearby. As users and their neighbors share similar preference, they perform check-ins to similar

venues. The venue aspect of spatial homophily says that venues that are near one another share more common features (e.g. visitors, rating) rather than between two venues that are far from each other.

Backstrom *et. al.* [4] studied spatial homophily in LBSNs using the Facebook Places of US users with known home city of users and their social connections. From this dataset, they showed that the probability of friendship between two users follows the power law distribution. The authors combined social and spatial homophily by assuming that users live near to their friends. Therefore, they proposed a statistical model to make use the information of home location of friends to infer home location of users.

Hu *et. al.* [35] considered venue aspect spatial homophily in their empirical analysis research on Yelp data. They found that most venues have neighbors with short distance; the average rating of a business is weakly positively correlated with those of its neighbors; and this correlation is independent of the categories of venues and their neighbors. Then, they proposed a matrix factorization approach to predict the number of check-ins between users and venues and developed four models each considering a different set of features: *neighborhood influence*, *review content*, *category influence* and *popularity and geo-distance influence*. Only the last model incorporates distance effect.

Le Falher *et. al.* [47] also considered venue aspect spatial homophily but they generalized this idea for neighborhood in cities. Their experiment showed that set of venues that are geographically close to each other could be grouped as a neighborhood because of their feature similarity. They evaluated their idea by finding the top-k similar neighborhoods in other cities given a neighborhood in one city. They found that using Earth Mover's Distance (EMD) [72] outperformed other measures in searching similar neighborhoods.

Liu *et. al.* [60] included *venue influence* of distance effect in their study of users' check-ins. From Gowalla dataset, they found that (i) a venue and its nearest neighbors tend to have more common users, and (ii) venues inside

the same region attract users with similar preferences. Therefore, the authors study the similarity between two venues at two different levels: *instance-* and *region-* levels using matrix factorization. The former refers to the similarity between venues and their neighbors while the latter studies the influence of venues in the same region. In their experiment, they also use multiple methods to construct regions and find out that no matter which method they use, the incorporation of region-level similarity always outperforms baselines. This result underscores the crucial impact of regional information in predicting check-ins.

Gao *et. al.* [28] proposed the **gSCorr** model for check-in data by combining *social homophily* with the *user aspect* of *spatial homophily*. This model assumes that users' check-in behavior is affected by distance between users and other users; time of check-in and social influence. Specifically, they divided geo-social correlations into four groups: local/distant friends, and local/distant non-friends. They showed that there is a positive correlation between the number of new check-ins and the percentage of new venues that have been checked-in by users from each group. The authors also observed that neighbor information of users improves the check-in prediction accuracy since users and their neighbors share more common activities. The drawback of this work is that it did not consider the competition of venues and also venues grouped into areas.

Li *et. al.* [51] classified three types of friends: *social friends*, *neighboring friends* and *location friends*. The authors illustrated the impact of these kinds of friends to check-in behavior of users. Based on the check-ins of these types of friends, the authors select the potential candidate venues used matrix factorization approach to predict the check-ins of users.

2.2.4 Neighborhood Competition

Venues need to compete with their nearby ones in order to attract the visitation of users. From our survey, neighborhood competition effect has not gained much attention of researchers despite of its importance in modeling check-in behavior. Liu *et. al.* [59] incorporated the competition effect by deriving the popularity score of each venue which represents the competition of the venue with its surrounding neighbors. The authors assumed that the probability of observing check-ins between user i and venue j is proportional to the distance between user i and venue j, popularity of venue j, and the interest of user i to venue j. To explore the interest between users and venues, the authors adopted *Latent Dirichlet Allocation* model [8] and *Bayesian Non-negative Matrix Factorization* [75] to study the preferences of users and venues.

Chapter 3

Empirical Analysis

In this chapter, we conduct empirical analysis on LBSN datasets to study various behavioral effects to check-in activities. We first describe the datasets, and their construction. We then describe how the exact home locations of users are obtained for the purpose of studying distance effect and user-aspect spatial homophily. This is followed by analysis of different behavioral effects.

3.1 Dataset Statistics

To study behavioral effects on check-ins at the venue level, we need datasets that cover all check-ins on all venues within a geographic region by a set of users. We choose to analyze check-in data generated within one city to avoid issues related to movement across cities. There are not many publicly available datasets that meet these criteria. We therefore crawled a Foursquare dataset that consists of 1.11 millions check-ins by Singapore users who publish their check-ins in public Twitter stream between August 15, 2012 and June 3, 2013. As shown in Table 3.1, this dataset (denoted as **SG**) consists of 55,891 users and 75,346 venues. These users declare Singapore to be their profile location.

We also crawled another set of Foursquare data generated by Indonesian users from July 2014 to May 2015 with 575,298 check-ins by 51,658 users on

Table 3.1: Dataset Statistics

	SG	**H_SG**	**JK**	**H_JK**	**NYC**
# users	55,891	856	14,974	455	7,092
# venues	75,346	12,020	38,183	4,380	21,287
# check-in's	1.11M	63,777	119,618	9,557	138,067
# user-venue pairs with > 0 check-ins	541,588	28,298	81,188	5,422	102,960

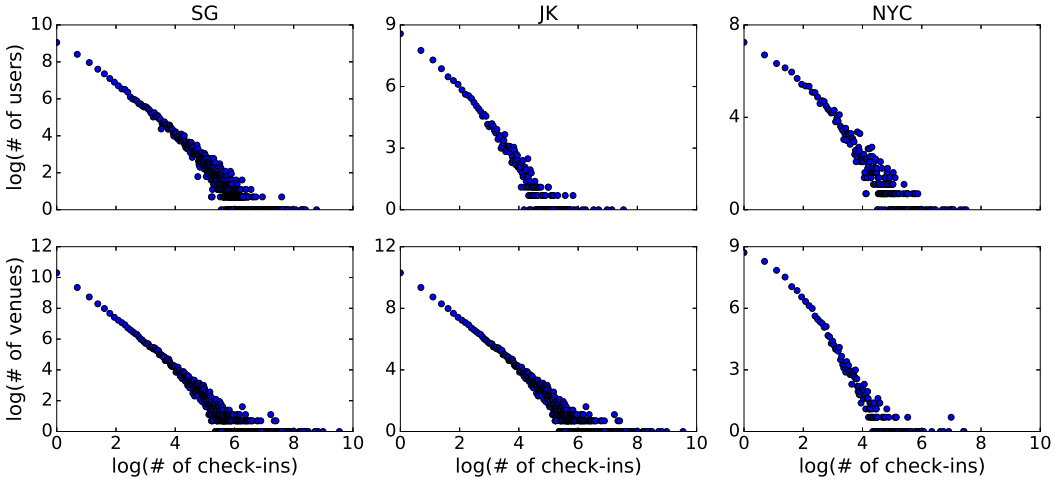


Figure 3.1: Distribution of check-ins over venues and users in **SG**, **JK** and **NYC** datasets.

216,847 venues [1]. These users declare Jakarta in their profile location. We further selected a subset of check-ins in Jakarta, the largest city in Indonesia. The statistics of this dataset (denoted as **JK**) is shown in Table 3.1. In the **JK** dataset, there are 119,618 check-ins performed by 14,974 users on 38,183 venues.

For more extensive evaluation, we also include the publicly available Gowalla [2] dataset [14]. The dataset contains all check-ins from February 2009 to October 2010. Since we only focus on check-ins within a city, we select check-ins of venues from New York City and denote them as **NYC**. As shown in Table 3.1, despite of having smaller number of users and venues, **NYC** still has more check-ins than **JK**. In other words, **NYC** is denser than **JK**.

[1]The dataset spans from 2 Aug 2011 to 13 May 2015 but there are only 99 check-ins from 2 August 2011 to end of June 2014 so we filter out this period.

[2]Gowalla is the location-based social network launched from 2007. It was reported to have 600,000 users on November 2010. After being acquired by Facebook on December 2011, it was closed in the beginning of 2012.

Moreover, **NYC** is also denser than **SG**.

Figure 3.1 provides the log scale of distribution of check-ins over users and venues in **SG**, **JK** and **NYC** datasets. All distributions have long tails which suggest that check-in distributions of users and venues are very skewed. In other words, very few users make large number of check-ins and very few venues receive large number of check-ins, but vast number of users perform one check-in only and vast number of venues receive only one check-ins.

3.2 Home Location Detection

Home location could influence a user's check-in behavior as part of several behavioral effects including distance effect and spatial homophily. For example, due to distance effect, people may prefer to visit supermarkets, attend schools and patronize fitness facilities in the home neighborhoods. If user-aspect of spatial homophily holds, users from the same neighborhood may be strongly correlated in their check-in behaviors. Unfortunately, **SG**, **JK** and **NYC** datasets do not provide information about the users' home locations required for analyzing the above behavioral effects.

In this research, we therefore select a subset of users whose home locations can be clearly identified using both their check-ins and check-in messages. Since we do not have additional information about venues (e.g. name of venues, reviews of users) in **NYC**, we cannot find the home location of users of **NYC**. The following are the detailed steps to identify the home locations of users in **SG** and **JK** datasets:

- We selected a subset of venues under the "home (private)" category which is in turn a sub-category of the "residence" category. This "home (private)" category is usually assigned to venues of home locations. In **SG**, there are 8,447 venues satisfying this criteria and 74,944 check-ins at these venues by 5,199 users. For **JK**, there are 7,800 checkins by 1,483

Table 3.2: Examples of "home" related key phrases to detect home locations in **SG** and **JK**.

Dataset	Keywords
SG	"back home", "home finally", "home sweet home", etc.
JK	"Tidur dulu"(sleep first), "Rumah"(House), "Pondok"(cottage), "sampai di rumah" (arrived to home), "bobo"(sleep), etc.

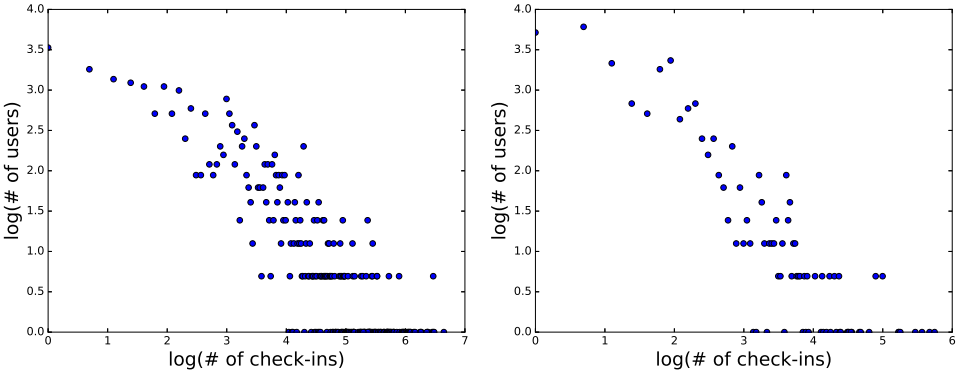
users at 1,985 venues. At this point, it is still unclear if these venues are the home locations of these users.

- We further selected a subset of 3,276 users in **SG** and 891 users in **JK** who have checked in at only one "home (private)" venue. This rules out users who have multiple "home (private)" venues.

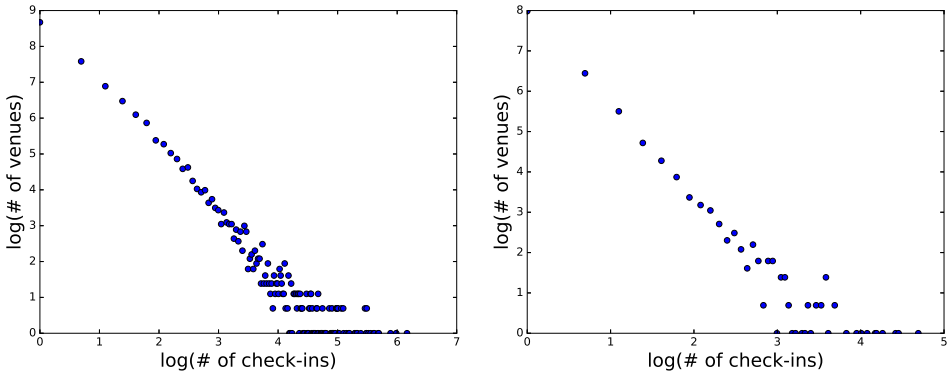
- We finally selected an even smaller set of users who also shouted some home relevant messages during their check-ins to the only "home (private)" venues. We use a set of "home" related key phrases to identify such messages in **SG** and **JK** datasets. Table 3.2 shows some examples of these key phrases in both datasets. As long as any of the key phrases is found, the check-in venue is used as the home location.

We finally obtained a dataset which includes 856 users and their home locations in **SG**. We call this dataset **H_SG**. These users have 63,777 check-ins on 12,020 venues as shown in Table 3.1. Note that this represents 1.5% of all users and 5.7% of all check-ins in **SG**. As a user can have multiple check-ins at the same venue, the number of unique user-venue pairs with non-zero check-ins is 28,298. Similarly, we obtained dataset **H_JK** for users in **JK** dataset. This subset has 455 users with 4380 venues and 9557 unique check-ins between them. These numbers correspond to 3% of users and 11.5% of venues in **JK** dataset. Moreover, there are 5422 user-venue pairs which have at least one check-ins between them.

Figure 3.2 shows the distributions of check-ins over users and venues in two datasets **H_SG** and **H_JK**. It is observed that these distributions follow power law distribution similar to datasets **SG** and **JK**. It suggests that our two



(a) **H_SG**. (b) **H_JK**.



(c) **H_SG**. (d) **H_JK**.

Figure 3.2: Distribution of check-ins over users and venues in **H_SG** and **H_JK**.

new datasets **H_SG** and **H_JK** still maintain the properties of their original datasets i.e. **SG** and **JK** respectively.

Since we cannot identify the home location of users in **NYC** dataset, we only use it in the analysis of *social homphily* and *neighborhood competition* effects.

3.3 Distance Effect

Previous works have shown that distance has an effect on the likelihood of a user visiting a venue [14, 22]. Some of these works studied the distance effect at the city level as only the city-level profile locations are available for most LBSN users [4, 54]. Others incorporated distance effect into their analysis or modeling works using the predicted home locations of users instead of user-

reported home locations [69, 14]. In the following, we conduct analysis using the distance between check-ins and actual home venues in **H_SG** and **H_JK** datasets. We want to examine the distance effect within a city, which has not been studied earlier.

For each user, we bin her check-ins according to the distance from the user's home location. Every 1-km distance range is a bin and we compute the probability of check-ins within each bin of a user by dividing the number check-ins within the bin by the total number of check-ins of this user. The average probability of check-ins of the distance bin is then the average of probabilities over all users. The maximum distance from home location to venue is 36.7 km in **H_SG** or 31 km in **H_JK**. As the large distance bins involve the check-ins of very few users, we exclude bins with distance larger than 26 km. As shown in Figure 3.3, the average probability of check-ins of distance bins further away from home location is smaller than that of distance bins nearer from home location. Hence, users are more likely to visit venues near their home locations rather than further away ones. The finding is consistent with other previous works [14, 54].

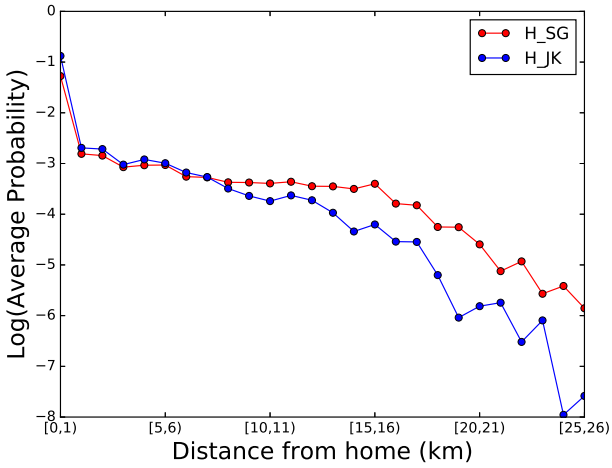


Figure 3.3: Fraction of check-ins as a function of distance from home in **H_SG** and **H_JK** datasets in log-scale. The base of log is 10.

Table 3.3: Average Jaccard scores between user-friend pairs versus random pairs of users across five datasets.

	SG	**H_SG**	**JK**	**H_JK**	**NYC**
Users and their friends	0.01411	0.01818	0.00697	0.01812	0.01921
Random pairs of users	0.00448	0.00867	0.00097	0.00085	0.00211

3.4 Social Homophily

Social homophily is the tendency that users and their friends share more common check-in venues than that between users and other ones. We illustrate the effect by calculating the average Jaccard similarity score of all pairs of users and their friends. Then, we compute the same score for equal number of random pairs of users. Specifically, each user u is represented by a set s_u containing all venues that u has visited and the Jaccard similarity of u and u' is $J(u, u') = \frac{|s_u \cap s_{u'}|}{|s_u \cup s_{u'}|}$.

Table 3.3 shows that the average Jaccard scores between users and their friends are significantly higher than that between random pairs of users. For example, the Jaccard score between users and their friends is three times higher than that of random user pairs in **SG** dataset. Moreover, the phenomenon is consistent across all the five datasets. Therefore, we could conclude that in LBSNs, users share more check-in venues with their friends than with strangers.

3.5 Spatial Homophily

3.5.1 User Aspect Spatial Homophily

When two users' home locations are near each other, there could be similarity between their check-ins due to the similar daily patterns shared by people living in the same neighborhood. This phenomenon is called the *user aspect* of *spatial homophily* which has been studied in some previous works. For instance, Li *et. al.* [51] combines this effect with social homophily to increase the performance of point-of-interest recommendation task. To detect the home

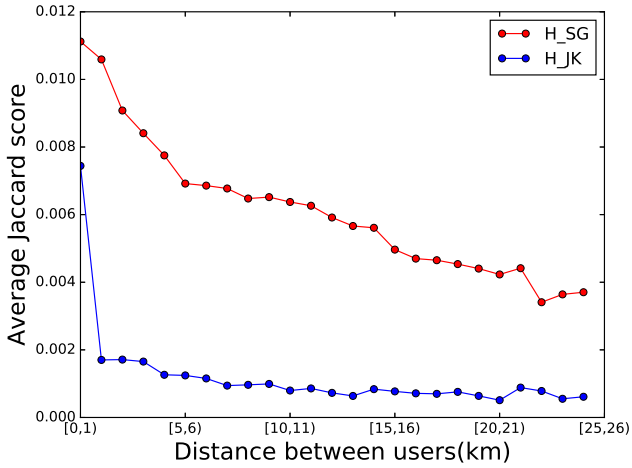


Figure 3.4: Relationship between Jaccard score and distance between every users in **H_SG** and **H_JK**.

location of users, they discretized the world into 25 by 25km cells and use the average position of check-ins in the most check-ins cells as home locations of users.

In this analysis, we use only the **H_SG** and **H_JK** datasets which contain user home location information. Figure 3.4 shows the average Jaccard similarity of check-in venues between pairs of users with different inter-home distance in both datasets. We first calculate the inter-home distance and Jaccard Similarity of check-in venues of every pair of users in **H_SG**. We then group pairs of users into distance bin of 1 km. For example, the first bin contains all user pairs whose distance is less than 1 km. The second bin contains user pairs whose distance is greater than 1 km and less than 2 km. We exclude those user pairs with distance larger than 26 km as they are few in number. We apply the same procedure to **H_JK**. Figure 3.4 shows that the average Jaccard Similarity decreases when the inter-home distance increases. Hence, neighbors are more likely to share common venues.

3.5.2 Venue Aspect Spatial Homophily

Using **H_SG** and **H_JK** datasets again, we want to examine the *spatial homophily* between venues and their neighbors. We investigate the visitor overlap

between a pair of venues over the distance between them to explore *spatial homophily*. We expect that the shorter the distance between two venues, the higher the visitor overlap between them. It is the indicator for *spatial homophily*. Specifically, for each venue j, we define a vector v_j of dimension size equal to the number of users in the dataset. Each element of v_j represents the interaction between a corresponding user and venue j. We introduce two definitions of v_j. The first definition assign the i-th element of v_j to the number of check-ins of user i performed at venue j. In other words, v_j contains the number of check-ins of every user to venue j. The second definition is that the i-th element of v_j is the distance from user i to venue j.

Before calculating the *cosine similarity* between every pair of venues, we divide the distance of venue pairs into bins of 1km width. For example, the i-th bin covers distance range between $i-1$ and i km. We exclude all venue pairs whose distance between them is greater than 31 km in both datasets due to the sparsity of such venue pairs. The average cosine similarity of all venue pairs whose distance within the bin is calculated and reported. The *cosine similarity* of a venue pair (j, k) is calculated by the following formula

$$\frac{(v_j \circ I^{jk}) \bullet (v_k \circ I^{jk})}{\|v_j \circ I^{jk}\| \|v_k \circ I^{jk}\|} \tag{3.1}$$

where $\circ$ and $\bullet$ are Hadamard and inner products of vectors respectively. I^{jk} is the binary vector of dimension size equal to number of users. Its i-th element equals to 1 if user i performs to both venues j and k; otherwise, the element equals to 0. Since we have two version of v_j, there are two corresponding *cosine similarities*: *distance cosine* and *check-in cosine*.

Figure 3.5 depicts the *cosine similarity* of all venue pairs for different distance bins of **H_SG** and **H_JK**. We observe from both datasets that: (i) the similarity between a pair of venues decreases if the distance between them increases, (ii) the trends are consistent regarding datasets or types of *cosine similarity* and (iii) despite of having the same trend, *distance cosine* and *check-*

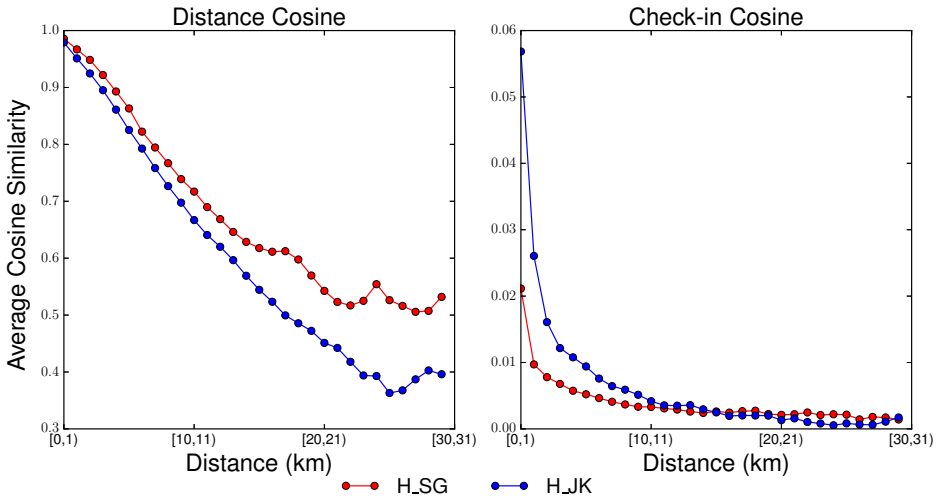


Figure 3.5: Spatial homophily through cosine similarity of all venue pairs over their distance in **H_SG** and **H_JK**.

in cosine have different shapes. While the former is nearly linear, the latter one follows a log-series distribution. In latter chapter, we will formalize *spatial homophily* by the two types of similarities.

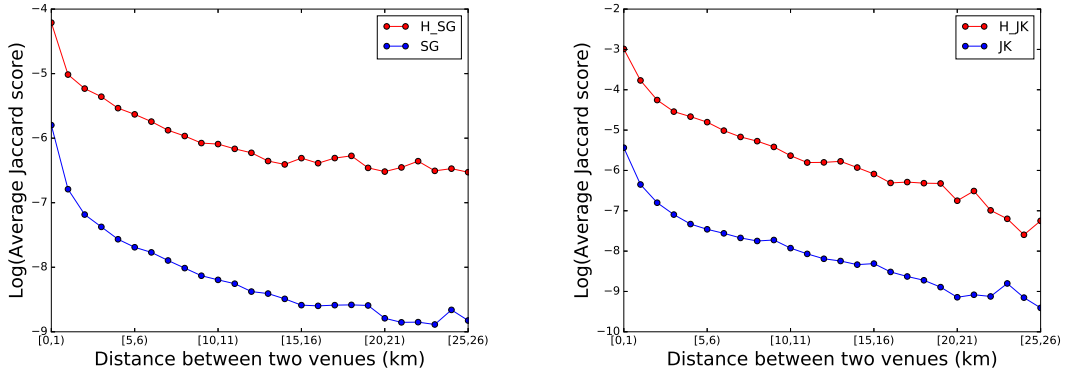
We also calculate the average Jaccard similarity of check-in users between pairs of venues separated from each other with different distances in our four datasets.

- Firstly, we calculate the distance and Jaccard similarity score of visited users of every pair of venues in our four datasets: **SG**, **H_SG**, **JK** and **H_JK**.

- Secondly, we group pairs of venues into distance bin of 1km and then calculate the average Jaccard similarity score of every pairs of venues inside each bin. We exclude the pairs whose distance is greater than 26km because the average Jaccard score of these pairs is equal to 0.

Figure 3.6 displays the average Jaccard similarity score of visited users between every pairs of the four datasets in log scale. From the figure, we observe that (i) all four datasets share the same trend, (ii) the average Jaccard score decreases when the distance of venue pair increases, and (iii) the trend of the four datasets follows power law distribution. Thus, Figure 3.6 suggests that

venues and their nearby neighbors tend to share more common users rather than venues that are far apart. In other words, *spatial homophily* exists among venues.



(a) **Singapore**. (b) **Jakarta**.

Figure 3.6: Relationship between average Jaccard score in log scale and distance between every pair of venues in **SG**, **H_SG**, **JK** and **H_JK**.

3.6 Area Attraction

Despite the distance effect, some venues may still attract check-ins from users far away. Li *et. al.* [54] developed an influence scope model for measuring the attractiveness of venues to their followers. In our research, instead of examining attractiveness at the venue level, we model attractiveness at the area level. There are three significant advantages of doing so. Firstly, it reduces the number of parameters in modeling which in turn reduces the learning time. Secondly, we address data sparsity issue at the venue level. Finally, we believe that the area a venue belongs to has a major influence over its ability to attract users. We are going to illustrate this by the following empirical analysis on only **H_SG** and **H_JK** datasets.

We empirically select three well known fast food chains, i.e., McDonald, KFC and Starbucks, with many branches. We expect branches of the same chain to be very similar to one another by food variety, food quality, ambience and service. Hence, at the venue level, we should not expect any difference

among their abilities to attract users from other locations. We now divide the city into non-overlapping square areas of width equals to 0.05 degree (the area width is equivalent to about 5.55 km on the equator) and assign every venue to exactly one area. Each area is assigned a center of the mass derived from the locations of its venues. We call the top five areas with most number of venues the *dense areas* while the areas from ranks 10 to 15 the *sparse areas*. We exclude other lower ranked areas as they do not contain any of the three fast food venues.

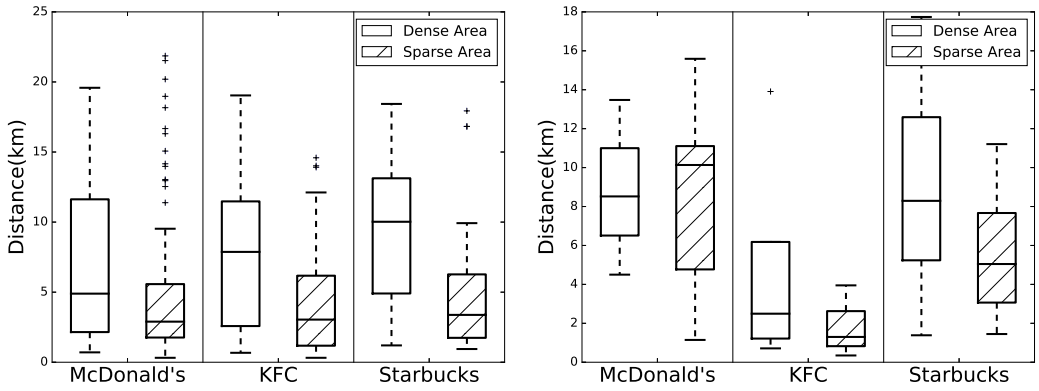
Table 3.4: The number of fast food stores in **H_SG** and **H_JK** datasets.

	McDonald	KFC	Starbucks
H_SG	108	89	95
H_JK	37	101	94

For each fast food chain, we examine the distances between each dense area (represented by its center of mass) and the home locations of users who perform check-ins to its venues inside the area. We then generate a boxplot for the user-area distance of all dense areas. We perform the same procedure for sparse areas. Figure 3.7 shows that for each fast food chain, branches within the *dense areas* attract users farther than branches in the *sparse areas*. This suggests that the attractiveness of area plays an important role bringing far away users to the venues in the area.

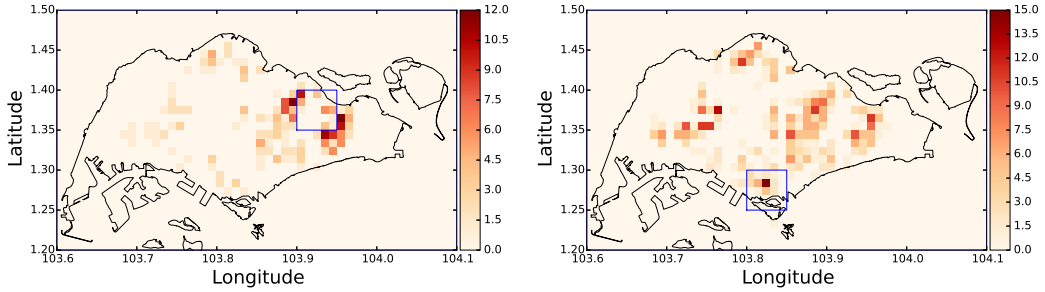
In Figure 3.7, there is an exception of McDonald chain in **H_JK** dataset. It could be explained that the number of stores of McDonald in **H_JK** is three times less than the one in **H_SG** than Jakarta users have to travel further to the McDonald outlets. The number of Starbuck and KFC outlets in **H_JK** and **H_SG** are quite similar (Table 3.4).

Figure 3.8 shows the case study of two areas and the location of their visitors. First of all, we divide a city into square areas and assign every venue to exactly one area. In Figures 3.8a and 3.8b, we show two areas, a_1 and a_2, covering the residential and downtown area respectively. Figure 3.8a shows



(a) **H_SG** (b) **H_JK**

Figure 3.7: Boxplot of distance from areas containing fast food chain to their check-ins users in **H_SG** and **H_JK**.



(a) Normal Resident Area a_1. (b) Downtown Area a_2.

Figure 3.8: Heatmap of number of users who make check-ins to different areas (blue square) in **H_SG** over map of Singapore.

that the users checking into area a_1 are largely from nearby areas. In contrast, Figure 3.8b shows users checking into a_2 can be from areas far away. This illustrates that area a_2 is more attractive than area a_1. In sequential chapter, we show that the attractive scores of areas are different so the potential to attract users of areas is also divergent.

3.7 Neighborhood Competition

To show competition among venues within the same area, we adopt the method originally proposed by Weng *et al.* [83] to study competition among memes in social networks. We divide the check-in history into weeks. We then measure

the following entropies for each week. During the measurement, some notations and their meaning are listed in Table 3.5.

Table 3.5: Table of Notation in Neighborhood Competition.

Notation	**Meaning**
$cks(v,t)$	number of check-ins of venue v within week t
$cks(u,t)$	number of check-ins of user u within week t
$cks(u,v,t)$	number of check-ins between user u and venue v within week t
A	set of all areas

- **System entropy (E_s):** $E_s(t) = -\sum_v f_v(t) \log f_v(t)$ where $f_v(t)$ is the fraction of check-ins in week t performed on venue v, i.e., $f_v(t) = \frac{\#cks(v,t)}{\sum_v \#cks(v,t)}$. The system entropy essentially measures the degree to which the distribution of check-ins concentrates on a small fraction of venues.

- **Average area entropy (E_A):** We define the entropy of an area a within week t to be $E_a(t) = -\sum_{v \epsilon a} f_{v,a}(t) \log f_{v,a}(t)$ and $f_{v,a}(t) = \frac{\#cks(v,t)}{\sum_{v \epsilon a} \#cks(v,t)}$. We then take the average of all area entropies, i.e., $E_A(t) = \frac{1}{|A|} \sum_{a \epsilon A} E_a(t)$. We divide the city into squares of 0.05 degree width. The construction of areas is discussed further in Chapter 5. Similar to system entropy, average area entropy captures the degree to which the distribution of check-ins of an area concentrates on a small fraction of venues (in the area).

- **Average user entropy (E_U):** We next define the average user entropy within week t as $E_U(t) = Avg_{u \epsilon U} E_u(t)$ where entropy of user u is $E_u(t) = -\sum_v f_{u,v}(t) \log f_{u,v}(t)$ and $f_{u,v}(t) = \frac{\#cks(u,v,t)}{\#cks(u,t)}$. This entropy quantifies the concentration of users' attention on the venues they perform check-ins on.

Figures 3.9 and 3.10 show the three entropies over weeks in the five datasets. The first observation is that both datasets show similar trends of the entropies.

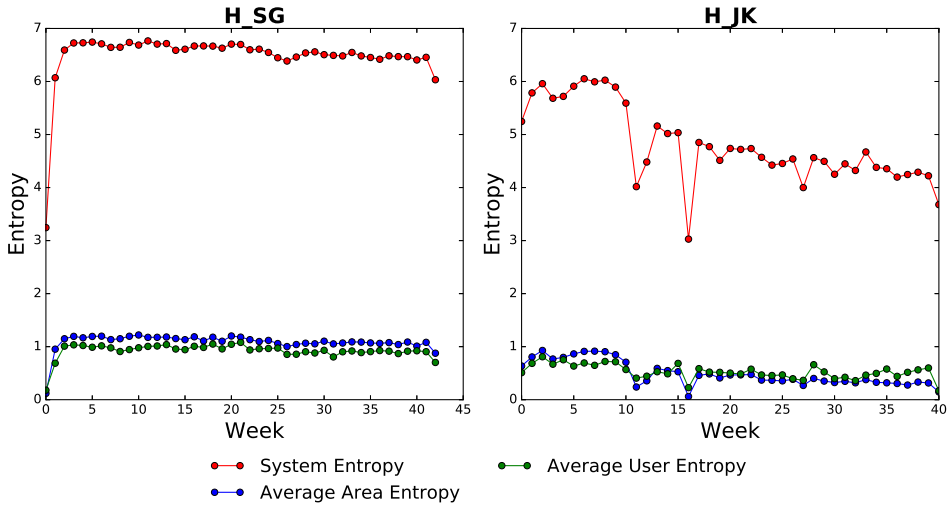


Figure 3.9: Weekly entropy in **H_SG** and **H_JK** datasets.

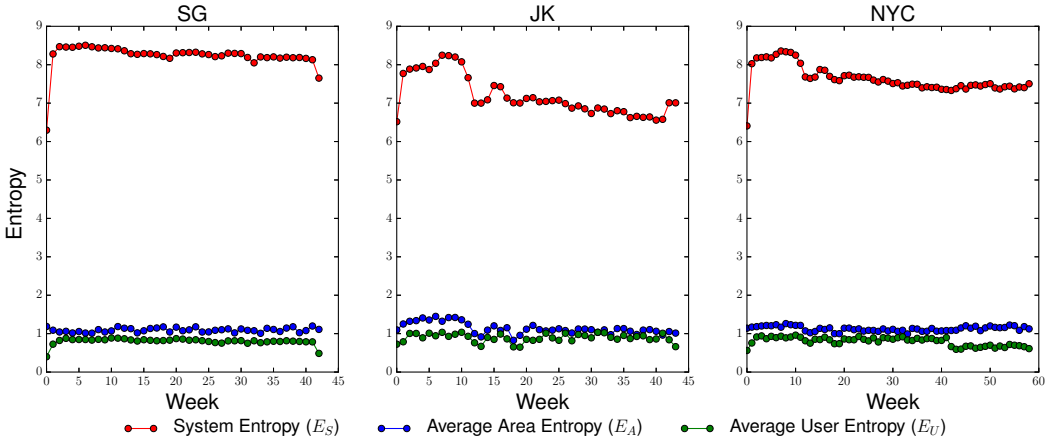


Figure 3.10: Weekly entropy in **SG**, **JK** and **NYC** datasets.

Secondly, the average user entropy is much smaller than system entropy. It clearly suggests that each user's attention is limited to very small fraction of venues in the entire city. Venues therefore have to compete to gain attraction from users. Thirdly, we observed from Figures 3.9 and 3.10 that system entropy is much larger than average area entropy across five datasets. This implies that check-ins within an area concentrated on smaller fraction of venues than the fraction of venues in the entire city receiving check-ins from the whole user population.

The above empirical analysis concludes that venues compete more with their nearby neighbors than those farther away. Thus, grouping venues into areas and modeling competition among venues in each area is therefore an

appropriate modeling approach.

3.8 Chapter Summary

In this chapter, we have conducted several empirical analysis on LBSN datasets in order to inspect the movement behavior of users. Firstly, we described the way of constructing our LBSN datasets. We later design empirical studies to illustrate different effects on user visitation. Specially, we have showed that *neighborhood competition* and *area attraction* are important effects which affect the visitation of users. Therefore, the following chapters are devoted to propose different models to study these two effects to understand the check-in behavior of users in LBSNs.

Chapter 4

PageRank-based Modeling of Venue Competition

In this section, we will propose a novel framework to measure the competitiveness of venues in LBSNs. Unlike popularity, competitive venues are expected to earn check-ins from their neighbors. For this reason, we construct the proximity graph of venues and turn visitation of users to venues into transition matrix which is then used to compute competitiveness of venues as their PageRank scores [64]. Moreover, we evaluate multiple configurations of our proposed model to investigate its robustness [21].

4.1 Overview of Venue Competitiveness Ranking

In this section, we describe some basic methods for ranking venues by competitiveness. We also highlight the strengths and weaknesses of these methods before presenting our proposed PageRank-based model.

- Popularity (e.g. Check-in count): This method ranks venues based on their number of check-ins. The more check-ins the venue has, the higher rank it has. Its advantage is that it is simple but it does not capture

the neighborhood competition of venues. For example, a venue v_a may enjoy very high popularity but does not have a single neighboring venue to compete with. On the other hand, another equally popular venue v_b at another location manages to compete with many neighboring venues to win lion share of check-ins. In this case, it is reasonable to rate v_b more competitive than v_a.

- Venue Influence: Li *et. al.* [54] proposed a method called *UDI* to rank venues based on the influence of venues to users. Specifically, the influence of a venue is high if it could attract users who live further away from its location. In other words, *UDI* assumes that a venue's competitiveness is the influence the venue has on its visitors who have to overcome distance effect on their check-in behavior. Again, this method does not involve any competition with the venue's neighbors.

Due to the lack of research works on neighborhood competition, we propose a PageRank-based model to derive a *venue's competitiveness* by its potential to win over its neighbors the visitation of users. The larger the *venue competitiveness* of venue, the higher chance for it to win visitation of users. Finally, we could use *venue competitiveness* to rank venues in LBSNs.

4.2 Proposed Venue Ranking Models

4.2.1 Overview of Ranking Framework

Before we present our proposed model, we first describe our overall framework to rank business venues using check-in data. We make two important assumptions. The first assumption is that the venues to be ranked are of the same type. Otherwise, it is not likely that the venues will compete with one another. In this work, while we do not consider area attractiveness, we also assume that competitions only occur between venues that are near each other.

Our proposed venue ranking framework consists of the following major steps:

- **Step 1: Construction of venue adjacency graph**: We first construct an undirected graph G consisting of venues as vertices. Two venues i and j are connected by an edge (i, j) if the distance between i and j is not more than λ, a distance threshold.

- **Step 2: Computation of venue competitive probability values**: Depending on the assumption used for modeling competitions among venues in winning check-ins from users, different venue competitive probability definitions p_{ji}'s can be worked out for the edges in G. We will elaborate these different definitions in Section 4.2.2.

- **Step 3: Computation of venue ranks**: In this step, we apply some PageRank-style models on the venue competitive probability values. The end results are venue ranks.

In the following, we shall elaborate the details of Steps 2 and 3.

4.2.2 Modeling Venue Competitive Probability

Given a venue adjacency graph with venues as nodes, we want to derive the competitive probability from one node j to another node i based on how much the venue value of j could be "distributed" (or lost) to i. Suppose i and j are in competition of some candidate users, the more users visiting i would suggest that the more j is losing the competition. Ideally, we would like to know: (a) the set of users considering to visit venue j, and (b) the subset of them actually visiting venue i instead. In most practical settings, we may observe (b) but not (a) unless the users are explicitly required to state their venue preferences. Without infringing the user private preferences, we would like to infer (a) using already observed visit data. In the following, we present

three approaches to derive competitive probability from one node to another using different assumptions.

Equal probability (EPR) assumption. Suppose venue j has $deg(j)$ neighboring venues. Without referring to any observed visit data, we assume that every neighboring venue of j will get equal share of visits. Let p_{ji} denote the competitive probability from venue j to venue i. We define $p_{EPR}(j,i)$ under the equal probability assumption as:

$$p_{EPR}(j,i) = \frac{1}{deg(j)} \tag{4.1}$$

Neighborhood check-in ratio (NCR) assumption. Suppose n_i denote the number of check-ins for any venue i. The neighborhood check-in ratio assumption states that the set of potential visits to a venue j is the sum of observed visits to j and its neighboring venues. Hence, under the NCR assumption, the competitive probability from venue j to venue i is defined as:

$$p_{NCR}(j,i) = \frac{n_i}{\sum_{j \leftrightarrow k} n_k + n_j} \tag{4.2}$$

where $j \leftrightarrow k$ denotes that j is a neighbor of k. The denominator $\sum\limits_{j \leftrightarrow k} n_k + n_j$ is essentially the sum of all check-ins observed on j and its neighbors.

Neighborhood user ratio (NUR) assumption. Suppose m_i denote the number of users performing check-ins on any venue i. The neighborhood user ratio assumption states that the set of potential users to a venue j is the sum of observed users to j and its neighboring venues. Hence, under the NUR assumption, the competitive probability from venue j to venue i is defined as:

$$p_{NUR}(j,i) = \frac{m_i}{\sum_{j \leftrightarrow k} m_k + m_j} \tag{4.3}$$

Next, we will apply the above competitive probability definitions to a few PageRank-style models that compute venue values.

4.2.3 PageRank Model

PageRank [64] was originally designed to compute the importance of web pages based on the directed links among the pages. The key idea of PageRank is that an important page should be linked from other important pages.

In our context, we define the first PageRank-style model with the competitive probabilities derived by the equal probability assumption. Let $PR_{EPR}(i)$ denote the value of venue i and is defined as:

$$PR_{EPR}(i) = (1-\alpha) \cdot \frac{1}{N} + \alpha \cdot \sum_{j \leftrightarrow i} PR_{EPR}(j) \cdot p_{EPR}(j,i) \tag{4.4}$$

where α is called *damping factor* to control the weight given to random walk in the PageRank calculation. In our experiments, we set $\alpha = 0.85$ by default. N denotes the total number of venues.

Given that we have two other competitive probability definitions, namely p_{NCR}, and p_{NUR}, the other two variants of PageRank Models can be derived, i.e., PR_{NCR}, and PR_{NUR} respectively.

4.2.4 CompetitiveRank Model

Other than the definition of competitive probability, we also explore other variants of PageRank style models by changing the random visits to any venues in the adjacency graph. In the PR_X models (where X denotes one of EPR, NCR, and NUR), every venue is visited with an equal probably $\frac{1}{N}$. This random visit scheme can be modified to create a hybrid PageRank-style model incorporating the observed visit data.

The new PageRank style model, known as **CompetitiveRank (CR)**, aims to combine the earlier PageRank models and the observed check-in data. We define the CompetitiveRank model in Equation 4.4.

$$CR_X(i) = (1-\alpha) \cdot \frac{n_i}{\sum_k n_k} + \alpha \cdot \sum_{j \leftrightarrow i} CR_X(j) \cdot p_X(j,i) \tag{4.5}$$

where X denotes one of EPR, NCR and NUR.

By varying the α parameter, we can moderate the effect of *check-in ratio* $\frac{n_i}{\sum_k n_k}$ of venue i, relative to the random walk effect. When $\alpha = 0$, CR_X reduces to check-in ratio.

4.3 Experiments on Real Datasets

In this section, we evaluate the proposed models using some real datasets collected from Foursquare. Our experiments consist of three steps. First of all, we will examine the correlations between models using Jaccard coefficient scores and Spearman correlation scores. Secondly, we evaluate the characteristics of the models by varying the distance parameter settings. Thirdly, we study a few case examples to show the difference between check-in count and PageRank style model. Finally, we evaluate the proposed models by comparing with Foursquare scores and number of likes of users to show the effectiveness of our methods.

4.3.1 Datasets

We collected Foursquare data during the period from 15 Aug 2012 to 3 June 2013 via Twitter. The data collected include check-ins of 55,891 Singapore users who have their check-ins posted as public tweets in their Twitter timelines. There are more than 1.64 millions check-ins at locations under different categories including building, food, and school. In our experiments, we only extract venues that are restaurants and their check-ins. We assume that these restaurants have to compete with other restaurants nearby. There are 121,439 check-ins at 7,290 restaurants in Singapore. For the ease of reading, we denote this dataset as $\mathbf{SG}_r$. Figure 4.1 summarizes the statistics of the dataset.

To determine a suitable distance threshold λ for defining the neighborhood of a restaurant, we plot the distribution of the distance between restaurants

Table 4.1: $\mathbf{SG}_r$ dataset statistics

# users	# check-ins	# restaurants	# check-ins of restaurants
55,891	1.64 millions	7,290	121,439

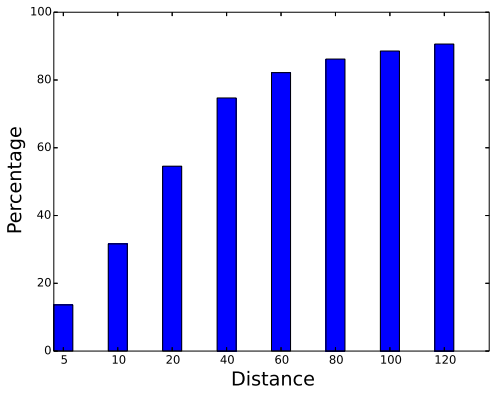
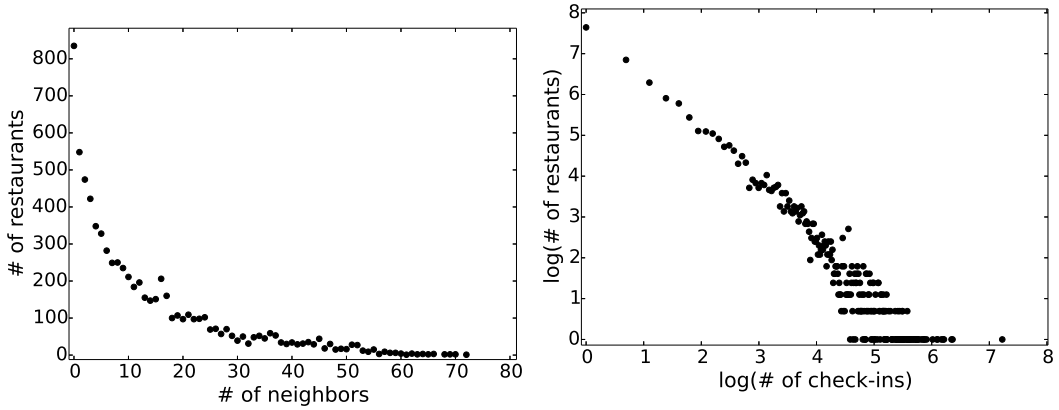


Figure 4.1: Proportion of restaurants with nearest neighbor distance $< x$ meters

and their nearest neighbors as shown in Figure 4.1. The figure shows that less than 12% of the restaurants have their nearest neighbors more than 100 meters away. This is not a surprise given that the city of Singapore is densely populated with food-related venues. We therefore set λ to be 100 meters to construct the network of restaurants.



(a) Degree distribution of restaurants.

(b) Check-in distribution of restaurants.

Figure 4.2: Distribution of restaurants in $\mathbf{SG}_r$ dataset.

Figure 4.2a shows the distribution of neighbor counts (degree distribution) of this restaurant network. The distribution has the log shape with large number of restaurants with a few neighbors and a few restaurants having large

number (as many as 50+) of neighbors. Besides, there are 835 restaurants which do not have any neighbors.

Figure 4.2b depicts the check-ins distribution of restaurant network. The figure shows that many restaurants have very few check-ins while few have many check-ins. The restaurants with the largest number of check-ins received 1,373 check-ins while 2,078 restaurants have only one check-in each.

We also apply our proposed models to two larger datasets: **SG** and **JK** and compare the results on them with the ones of restaurant dataset. Both **SG** and **JK** have been described in Chapter 3. The parameters of models are similar to the ones of restaurant dataset. Particularly, λ = 100 meters and α = 0.85.

4.3.2 Correlation Analysis

We have altogether six different PageRank style models for determining venue values and they are based on different competitive probability definitions and random visit options. The first part of the experiment thus seeks to determine how different they are when applied on our real dataset using correlation analysis.

We evaluate the models' correlation using (i) *Jaccard Coefficient at Top-k* venues, and (ii) *Spearman correlation coefficient*. The Jaccard Coefficient of two sets X and Y is defined by $\frac{|X \cap Y|}{|X \cup Y|}$. By considering the top k ranked venues returned by each model, we derive the Jaccard Coefficient of the top k ranked venues. Instead of using any k values, we consider $k = 100, 200$, and 300 to focus on overlaps among top ranked venues.

For a set of N venues with venue i assigned with ranks x_i and y_i by models A and B respectively, the Spearman correlation coefficient is defined as $1 - \frac{6 \sum d_i^2}{N(N^2-1)}$ where $d_i = x_i - y_i$. Venues with rank tie are assigned the average rank position.

Table 4.2 shows the Jaccard Coefficients of different pairs of models for

Table 4.2: Jaccard Coefficient@top k of $\mathbf{SG}_r$, **SG** and **JK** datasets. All models have $\alpha = 0.85$. All Jaccard coefficient scores greater than 75% are in bold text. The unit in table is percentage.

		PR_{NCR}			PR_{NUR}			CR_{EPR}			CR_{NCR}			CR_{NUR}		
	Top K	100	200	300	100	200	300	100	200	300	100	200	300	100	200	300
$\mathbf{SG}_r$	PR_{EPR}	0.5	5	6.2	1	4.4	5.8	0.0	3.1	4.3	0.0	1.5	3.3	0.0	1.8	2.9
	PR_{NCR}	-	-	-	70.9	**79.4**	**79.6**	10.5	12.4	15.6	30.7	36	41.2	27.4	33	39
	PR_{NUR}	-	-	-	-	-	-	8.7	13	14.9	23.5	35.1	39	21.2	35.6	39.5
	CR_{EPR}	-	-	-	-	-	-	-	-	-	39	37.9	39.8	39.8	37.9	40.2
	CR_{NCR}	-	-	-	-	-	-	-	-	-	-	-	-	**81.8**	**85.2**	**89.9**
SG	PR_{EPR}	0.0	0.5	1.01	0.0	1.01	1.35	0.0	0.5	0.67	0.0	0.5	0.84	0.0	0.5	0.84
	PR_{NCR}	-	-	-	**78.5**	61.9	63.1	39.9	33.8	34.2	53.8	55.6	53.8	52.6	50.4	46.3
	PR_{NUR}	-	-	-	-	-	-	40.8	30.7	32.2	53.8	50.3	47.7	57.4	51.5	47.7
	CR_{EPR}	-	-	-	-	-	-	-	-	-	65.3	48.7	51.1	58.7	48.2	49.6
	CR_{NCR}	-	-	-	-	-	-	-	-	-	-	-	-	**77**	**82.7**	**80.2**
JK	PR_{EPR}	0.0	0.0	0.3	0.0	0.0	0.8	0.5	0.3	0.3	0.0	0.0	0.2	0.0	0.3	0.7
	PR_{NCR}	-	-	-	72.4	63.3	58.3	25.0	21.2	23.0	48.1	49.8	54.2	46.0	45.5	46.7
	PR_{NUR}	-	-	-	-	-	-	23.5	19.0	17.6	41.8	40.4	42.5	50.4	50.4	55.0
	CR_{EPR}	-	-	-	-	-	-	-	-	-	46.0	40.8	40.2	42.9	35.6	33.0
	CR_{NCR}	-	-	-	-	-	-	-	-	-	-	-	-	63.9	62.6	61.7

different top k's. Generally, PR_{EPR} model is most different from the other models. CR_{EPR} is also different from other models but is more similar to other CR models than PR_{EPR} and other PR models. The most similar model pairs however go to the (PR_{NCR}, PR_{NUR}) and (CR_{NCR}, CR_{NUR}) pairs. These two pairs of models enjoy more than 70% overlaps between their top k ranked venue venues. The difference between PR and CR models can be explained by the damping factor. In CR model, it is usually larger than PR's one because the number of venues is smaller than the number of check-ins.

Table 4.2 also shows the results for the two datasets **SG** and **JK**. From the table, we can draw some observations. Firstly, the overlaps between two pairs (PR_{NUR}, PR_{NCR}) and (CR_{NUR}, CR_{NCR}) are higher than other pairs (e.g. (PR_{EPR}, PR_{NCR}). This observation is clearer in **SG** dataset than in **JK** since the Jaccard scores of the two pairs are greater than 60%. Secondly, the Jaccard scores of **SG** and **JK** are quite consistent. For example, both datasets see PR_{EPR} completely different from other models because of the low Jaccard scores. Lastly, the result from Table 4.2 is consistent with the results using $\mathbf{SG}_r$ dataset.

Table 4.3: Spearman correlation coefficient of $\mathbf{SG}_r$, **SG** and **JK** datasets. Coefficients greater than 0.70 are boldfaced.

		PR_{NCR}	PR_{NUR}	CR_{EPR}	CR_{NCR}	CR_{NUR}
$\mathbf{SG}_r$	PR_{EPR}	0.15	0.16	0.29	0.228	0.23
	PR_{NCR}	-	**0.96**	-0.0069	**0.73**	0.667
	PR_{NUR}	-	-	-0.0096	0.692	0.68
	CR_{EPR}	-	-	-	0.581	0.62
	CR_{NCR}	-	-	-	-	**0.974**
SG	PR_{EPR}	0.44	0.1	0.27	0.16	0.3
	PR_{NCR}	-	-0.12	0.44	-0.14	0.47
	PR_{NUR}	-	-	**0.77**	**0.8**	0.59
	CR_{EPR}	-	-	-	0.56	**0.83**
	CR_{NCR}	-	-	-	-	**0.71**
JK	PR_{EPR}	0.48	0.23	0.44	0.31	0.44
	PR_{NCR}	-	-0.08	0.55	-0.04	0.67
	PR_{NUR}	-	-	**0.7**	**0.83**	0.48
	CR_{EPR}	-	-	-	0.58	**0.85**
	CR_{NCR}	-	-	-	-	0.61

Now, we evaluate the Spearman rank correlation of the full rank lists returned by each pair of models as shown in Table 4.3. This allows us to answer the question whether the models are similar for their full rank lists. For the case of $\mathbf{SG}_r$ dataset, Table 4.3 essentially confirms that (PR_{NCR},PR_{NUR}) and (CR_{NCR},CR_{NUR}) model pairs are most similar. In fact, both model pairs enjoy > 0.9 correlation coefficient values. The result is consistent with that of Table 4.2. For the case of **SG** and **JK**, Table 4.3 for $\mathbf{SG}_r$ with one exception. The pair (PR_{NUR}, PR_{NCR}) produces two slightly opposite rankings since its Spearman score is negative in both datasets and it is different from that of $\mathbf{SG}_r$ when the pair shows the strong correlation. Since the rankings of top-k venues between them are very similar (see Table 4.2), it is clear that they generate different rankings for lower venues.

4.3.3 Case Examples

In this section, we show two case examples to illustrate how our proposed CR_{NUR} model differs from check-in count when ranking the venues. We include case examples using the CR_{NUR} model because it in general is similar to other models across the three datasets. Specifically, the similarity score

Table 4.4: Case Studies of Our Model in $\mathbf{SG}_r$ dataset.

Venues Name	# Check-in's (Rank)	CR_{NUR} (Rank)	# Neighbors	Avg CR_{NUR} of neighbors	Avg CR_{NUR} Rank of neighbors
Case study 1					
The Manhattan Fish Market	139 (136^{th})	0.0019 (39^{th})	42	0.00025	2682.31 th
Case study 2					
BALIthai	59 (494^{th})	0.00071 (298^{th})	55	-	2433.82 th
Xin Wang Hong Kong Cafe	130 (158^{th})	0.00068 (312^{th})	10	-	3149.6^{th}

of CR_{NUR} and CR_{NCR} is relatively high compared with that of other model pairs. Initially, we present two case studies of $\mathbf{SG}_r$ dataset and then two other cases in **SG** dataset. **JK** is not included in this analysis because of the lack of language knowledge.

Case Study 1 of $\mathbf{SG}_r$. The first part of Table 4.4 shows the *The Manhattan Fish Market* restaurant. The restaurant has about 139 check-ins, a high number compared to other restaurants. Hence, it is ranked 136^{th} according to check-in count. By CR_{NUR} model, however, *The Manhattan Fish Market* is ranked much higher at 39^{th} place. The result can be explained by the CR_{NUR} values of *The Manhattan Fish Market*'s neighbors. According to the Table 4.4, the average CR_{NUR} of *The Manhattan Fish Market*'s neighbors is high given the average rank 2682.31 is higher than the middle rank of $\frac{7890}{2} = 3945$.

Case Study 2 of $\mathbf{SG}_r$. The second part of Table 4.4 shows two venues *BALIthai* and *Xin Wang Hong Kong Cafe* that are ranked in different order by check-in count and by CR_{NUR}. By check-in count, *BALIthai* is ranked lower than *Xin Wang Hong Kong Cafe*. By CR_{NUR}, however, we have the reverse rank order due to the higher average CR_{NUR} rank of *BALIthai*'s neighbors. The better ranked neighbors suggest that *BALIthai* must be quite good so as to win visits from these neighboring competing venues. Moreover, the Foursquare score of *BALIthai* is 6.9 with 6 likes from users while *Xin Wang Hong Kong Cafe*'s score is 5.71 with 4 likes. This fact gives us more confident about the superior of CR_{NUR}.

Table 4.5: Case Studies of Our Model in **SG** dataset

Venues Name	# Check-in's (Rank)	CR_{NUR} (Rank)	# Neighbors	Avg CR_{NUR} of neighbors	Avg CR_{NUR} Rank of neighbors
Case study 1					
Carot Cake @264	1 (60465^{th})	4.06×10^{-5} (3670^{th})	22	5.481×10^{-5}	3018.54^{th}
Case study 2					
Widevision Asia	140 (1115^{th})	3.3×10^{-5} (4455^{th})	7	-	7567^{th}
Blk 310, Woodlands St31	13 (30912^{th})	3.6×10^{-5} (4043^{th})	17	-	3252.15^{th}

Case Studies 1 of SG. As shown in Table 4.5, the venue named *Carot Cake @264* has only one check-ins from users and its rank is 60465 based on number of check-ins. However, its rank using CR_{NUR} is 3670, significantly higher than the rank of number of check-ins. The reason can be explained by the ranks of its neighbors. According to Table 4.5, the average rank of its neighbors is 3018.54 which is higher than the average rank $\frac{75346}{2} = 37673$.

Case study 2 of SG Table 4.5 shows two different venues named *Widevision Asia* and *Blk 310, Woodlands St31* are ranked differently by check-in count and CR_{NUR}. According to check-in count, *Widevision Asia* has 10 times more check-ins than *Blk 310, Woodlands St31* but in CR_{NUR}, the latter one is ranked higher than the former one. The reason is that the neighbors of *Blk 310, Woodlands St31* have average rank higher than that of *Widevision Asia* so each winning of *Blk 310, Woodlands St31* is more valuable than that of *Widevision Asia*. Moreover, the number of neighbors of *Blk 310, Woodlands St31* is more than that of *Widevision Asia* so *Blk 310, Woodlands St31* can earn more score from each of its winning over the neighbors.

Although we only show CR_{NUR} in the above examples, there are many other similar case examples that we can extract from other PageRank models.

4.3.4 Evaluation with Foursquare Score Data.

Foursquare provides a score to each venue to reflect users' opinions about the venue by combining user's response such number of check-ins, number of likes,

Table 4.6: Top-k performance in $\mathbf{SG}_r$, **SG** and **JK** datasets.

	k	Check-in count	PR_{EPR}	PR_{NUR}	PR_{NCR}	CR_{EPR}	CR_{NUR}	CR_{NCR}
$\mathbf{SG}_r$	10	7.737	2.52	**8.071**	**8.081**	6.027	7.61	7.61
	20	6.749	2.405	**7.9325**	**7.9825**	5.942	6.8895	7.1865
	50	7.002	2.532	**7.11**	**7.0862**	6.2682	6.936	6.952
SG	10	6.872	2.327	**7.189**	**8.241**	5.836	6.89	5.721
	20	6.432	2.781	**7.625**	**7.715**	5.421	6.021	4.9652
	50	5.982	3.141	**6.91**	**6.843**	5.2	5.825	5.623
JK	10	7.145	3.197	**7.18**	**7.22**	5.122	7.034	5.62
	20	7.218	3.451	**7.5**	**7.369**	5.817	7.341	5.41
	50	7.155	3.048	**7.19**	**7.412**	5.821	7.16	5.1

Table 4.7: Spearman correlation of Foursquare score and all models in $\mathbf{SG}_r$, **SG** and **JK** datasets.

Dataset	Check-ins count	PR_{EPR}	PR_{NUR}	PR_{NCR}	CR_{EPR}	CR_{NUR}	CR_{NCR}
$\mathbf{SG}_r$	0.0476	-0.0488	0.1148	0.1358	-0.07	0.027	0.0417
SG	0.1639	-0.1091	0.1646	0.1977	0.0348	0.1288	0.1568
JK	0.1356	-0.0911	0.1366	0.1782	0.0145	0.1134	0.1298

and tips. The Foursquare score is between 0 and 10. Thus, we could use the Foursquare score to evaluate our models.

Table 4.6 shows the average Foursquare score of top k venues returned by each model in the three datasets. PR_{NUR} and PR_{NCR} are the winners as they have higher scores in three out of four cases. CR_{NUR} performs worse than PR_{NUR} and PR_{NCR} but its result is similar to the Check-in count.

Table 4.7 shows the Spearman correlation between the Foursquare scores and ranking scores of restaurants returned by the proposed models. CR_{EPR} and PR_{EPR} have negative correlation while PR_{NUR} and PR_{NCR} have strong positive correlation with Foursquare scores. CR_{NUR} and CR_{NCR} have positive correlation with Foursquare scores but the correlation is weak, in fact weaker than *Check-in count*. The results from Table 4.6 and Table 4.7 are consistent because both tables show the superior performance of PR_{NCR} and PR_{NUR} over the other models. These above observations are consistent across the three datasets.

Table 4.7 shows the Spearman correlation of Foursquare score and our ranking models in $\mathbf{SG}_r$, **SG** and **JK** datasets. As shown in the table, the

performance of all models are similar in the three datasets. Specifically, the performance of PR_{NUR} and PR_{NCR} are better than check-in count model since they are more similar to the ranking of Foursquare score. However, the improvements of PR_{NUR} over check-ins count model in **SG** and **JK** are less than the one of PR_{NUR} in $\mathbf{SG}_r$. The ranking of PR_{EPR} is negatively correlated to the ranking of check-in count. Secondly, Table 4.7 shows the same trending for the three datasets. The reason for the superiority of PR_{NUR} and PR_{NCR} is that these models consider the visitation of users to venues under the influence of venues nearby. This implicit property cannot be covered by check-in count model. In other words, neighborhood competition is an important effect which should be considered in modeling check-in behavior of users in LBSNs.

Chapter 5

Modeling Neighborhood Competition and Area Attractiveness in Check-in Behavior For Partially Known User Home Locations.

In this chapter, we propose the **V***isitation by* **A***ttractiveness and* **N***eighborhood Competition* (VAN) model for check-in behavior which incorporates area attractiveness, neighborhood competition and distance effects. Here, the home location information of users is assumed to be known. We further develop the parameter learning approach for *VAN* model and discuss its implementation.

Our experiments using synthetic and real datasets show that VAN model outperforms the baseline models for several tasks, including home location prediction for users with unknown home locations, venue competitiveness ranking and check-in prediction.

Table 5.1: Table of Notations for VAN model.

Notations	**Meaning**
U	set of all users
V	set of all venues
C	set of all check-ins
w_{iv}	number of check-ins of user i to venue v
w_v	total number of check-ins of venue v
a_v	area a_v containing venue v
s	the width of area
σ_v	competitiveness score of venue v
σ_{a_v}	attractiveness of area a_v
$N(v)$	set of neighboring venues of v
$i \to a_v$	user i visiting area a_v

5.1 Proposed Model

Let U and V denote the set of users and venues in a city respectively. We divide the city into mutually exclusive square areas of width s. We use a_v to denote the *area* which contains v. More notations and their meanings are shown in Table 5.1.

For each check-in between user and venue, the **VAN** model captures *area attraction*, *neighborhood competition* and *distance effects*. The VAN model adopts the following assumptions:

- Every user chooses an area to perform a check-in based on its attractiveness and the distance between the user and area.
- Every venue must compete against its neighboring venues in order to gain any check-in.

We assign each venue v a **competitiveness** value σ_v to measure its ability to compete with its neighbors. The value of σ_v is positive, and the larger the σ_v the more competitive the venue v.

There are multiple ways to define the **neighborhood** $N(v)$ of venue v but it should cover v and also the area containing v.

Every check-in of user i to venue v follows a two-step process. Firstly, user i must select the area a_v. Secondly, the venue v in area a_v must win over all

other neighboring venues in $N(v)$ to gain a check-in from user i.

- User i selects the area a_v under the effect of attractiveness of area a_v. Moreover, if the distance between i and a_v increases, the probability of user i chooses area a_v decreases. We model this by zero-mean Gaussian distribution whose variance is σ_{a_v}. The Euclidean distance between user i and a_v is the random variable which is generated from the distribution. In other words, the home location of user i is generated from the Gaussian distance whose mean is the location of area a_v and variance is σ_{a_v}.

- To model the winning of venue v over its neighbors, we need to model the difference of competitiveness of v and that of one of its neighbors, say v' . We propose two options. The first option uses the *cumulative distribution function (CDF)* of standard Gaussian distribution i.e. $CDF(\sigma_v - \sigma_{v'}; 0, 1)$. The second option is *Sigmoid function* of $\sigma_v - \sigma_{v'}$, i.e. $S(\sigma_v - \sigma_{v'})$. Both functions return probability values because they map differences between the competitiveness values of two venues into the range $[0, 1]$. If venue v is more competitive than its neighbor v' i.e. $\sigma_v > \sigma_{v'}$, the two functions will return a higher probability of v winning the check-in over v' and vice versa.

Formally, we consider the probability of a check-in from user i to venue v, p_{iv}, as follows:

$$p_{iv} = p(i \to a_v) \prod_{v' \in N(v)} p(v > v') \tag{5.1}$$

Equation 5.1 says that p_{iv} depends on two components: $p(i \to a_v)$ denoting the probability of user i selecting area a_v and $p(v > v')$ denoting the probability of venue v winning over its neighbors v'.

Let (x_i, y_i) and (x_{a_v}, y_{a_v}) denote the location of user i and center of area a_v respectively. Formally, the probability $p(i \to a_v)$ is defined by:

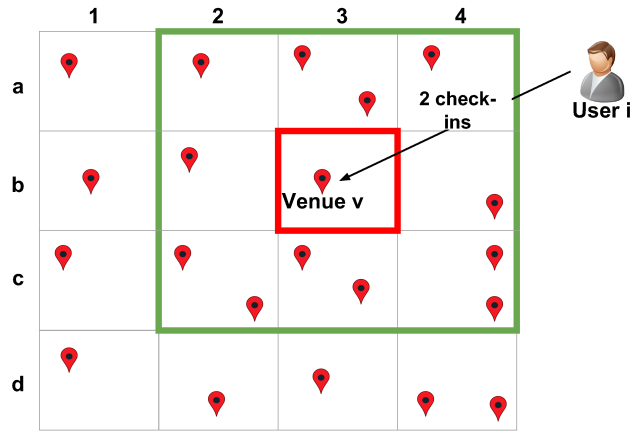


Figure 5.1: Example of Check-in graph.

$$
\begin{aligned}
p(i \to a_v) &= \mathcal{N}\left(\begin{pmatrix} x_i \\ y_i \end{pmatrix}; \begin{pmatrix} x_{a_v} \\ y_{a_v} \end{pmatrix}, \begin{pmatrix} \sigma_{a_v}^2 & 0 \\ 0 & \sigma_{a_v}^2 \end{pmatrix}\right) \\
&= \frac{1}{\sqrt{2\pi\sigma_{a_v}^2}} \exp\left(\frac{(x_i - x_{a_v})^2 + (y_i - y_{a_v})^2}{-2\sigma_{a_v}^2}\right)
\end{aligned} \tag{5.2}
$$

We model the attractiveness of each area a_v by a bivariate Gaussian distribution with center of area as the mean and covariance matrix representing the attractiveness of a_v, i.e., σ_{a_v}. The larger Euclidean distance between user i and center of area a_v, the smaller the $p(i \to a_v)$. The covariance matrix is diagonal and the diagonal elements share the same value σ_{a_v} because we assume that the attractiveness of area a_v in *x-axis* is similar to that in *y-axis*.

Neighborhood competition is modeled by the probability $p(v > v')$ of venue v winning a check-in over venue v' which can be defined by either a Sigmoid function or cumulative density function of standard Gaussian distribution. Formally,

$$
p(v > v') = \begin{cases} S(\sigma_v - \sigma_{v'}) & \text{if } VAN_{Sigmoid} \\ CDF(\sigma_v - \sigma_{v'}; 0, 1) & \text{if } VAN_{CDF} \end{cases} \tag{5.3}
$$

Depending on the choice of the above definitions, we have two variants of VAN models denoted by $\mathbf{VAN}_{Sigmoid}$ and $\mathbf{VAN}_{CDF}$.

Example: Figure 5.1 depicts two check-ins at venue v by user i i.e. $w_{iv} = 2$.

The **neighbors** of venue v, $N(v)$, are venues within area (b, 3) (red box) and its adjacent areas (i.e. boxes limited by green border). To perform a check-in at venue v, user i has to select area (b, 3) (enclosed by red box) considering the distance from his home location to the center of area (b, 3) and the attractiveness of area (b, 3). Moreover, the venue v needs to *win* over all of its neighbors in the adjacent areas (i.e. venues within the green box).

5.2 Inference

To learn the VAN model, we could use the standard technique Maximum Log-likelihood Estimation(MLE) but there is no closed form solution to find the global optima point. We instead propose a way to find local optimal points of this model.

The log-likelihood of a set of check-ins C from users from U on venues from V is then defined as:

$$\begin{aligned}
&\mathcal{L}(C|\{\sigma_v\}_{v\in V}) = \sum_{(i,v)\in C} w_{iv} \log p_{iv} \\
&= \sum_{(i,v)\in C} w_{iv} \log p(i \to a_v) + \sum_v w_v \sum_{v'\in N(v)} \log p(v > v') \\
&= \sum_{(i,v)\in C} w_{iv} \left(-2\log \sigma_{a_v} - \frac{1}{2\sigma_{a_v}^2}((x_i - x_{a_v})^2 + (y_i - y_{a_v})^2) \right) \\
&\quad + \sum_v w_v \sum_{v'\in N(v)} \log p(v > v') + const
\end{aligned} \tag{5.4}$$

In Equation 5.4, w_{iv} denotes the number of check-ins between user i and venue v, and w_v denotes the total number of check-ins on venue v.

Inference of user home locations: Taking derivative with respect to the x-coordinate of user i and set it to 0 gives us:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial x_i} &= \sum_v w_{iv} \left(-\frac{1}{2\sigma_{a_v}^2} 2(x_i - x_{a_v}) \right) = 0 \\
x_i &= \frac{\sum_v \frac{w_{iv}}{\sigma_{a_v}^2} x_{a_v}}{\sum_v \frac{w_{iv}}{\sigma_{a_v}^2}}
\end{aligned} \tag{5.5}$$

Similarly, we obtain the update function for y_i as:

$$y_i = \frac{\sum_v \frac{w_{iv}}{\sigma_{a_v}^2} y_{a_v}}{\sum_v \frac{w_{iv}}{\sigma_{a_v}^2}} \tag{5.6}$$

In Equations 5.5 and 5.6, we could not take x_{a_v} and y_{a_v} out of the sum because a_v is the area associated with venue v.

Based on Equations 5.5 and 5.6, we derive some interesting observations about the home location of user i.

- The home location of user i is the *weighted average* of centers of areas of venues a_v checked in by i.
- The weight $\frac{w_{iv}}{\sigma_{a_v}^2}$ associated to each area a_v has two components: w_{iv} the number of check-ins of user i to venue in the area and σ_{a_v} the attractiveness of the area. The former helps to predict the home location close to the check-in area due to distance effect. However, area attractiveness has an inverse effect on the importance of area. That is, more attractive areas should contribute less to identifying the home location of user i.
- Suppose the maximum and minimum values of x-coordinate (i.e. latitude) of city are x_{max} and x_{min} respectively i.e. $\forall a_v : x_{min} \leq x_{a_v} \leq x_{max}$, we have $\forall i \in U : x_{min} \leq x_i \leq x_{max}$. Similarly, if y_{max} and y_{min} are maximum and minimum values of y-coordinate (i.e. longitude) of city respectively, we have $\forall i \in U : y_{min} \leq y_i \leq y_{max}$. In other words, the weighted average of centers of check-in areas ensures that the home location of user i is within the city boundary.

Inference of competitiveness of venues: To maximize the log likelihood $\mathcal{L}$ with the respect to σ_v and the constraint $\sigma_v > 0$, we add the regularization term $\sum_{v \in V} \log \sigma_v$ and use *gradient descent* to find the optimal values of σ_v. The regularization term $\sum_{v \in V} \log \sigma_v$ keeps all σ_v values positive because if $\exists v \in V : \sigma_v \to 0$, $\log \sigma_v$ and $\sum_{v \in V} \log \sigma_v$ will become $-\infty$.

Formally, we have the optimization problem

$$\{\sigma_v^*\}_{\{v\}} = arg \max_{\forall v \in V: \sigma_v} \mathcal{L}(\{\sigma_v\}_{v \in V}) + \sum_{v \in V} \log \sigma_v \tag{5.7}$$

$\mathcal{L}(\{\sigma_v\}_{v \in V})$ denotes $\sum_{v \in V} \mathcal{L}(\sigma_v)$. We then define the log-likelihood for the σ_v of each venue v

$$\begin{aligned} & \mathcal{L}(\sigma_v) \\ = & \sum_{\substack{i \in U, \\ v' \in a_v}} w_{iv'} \log p(i \to a_v) + \sum_{\substack{i \in U, \\ v'' \in N(v) \setminus a_v}} w_{iv''} \log p(i \to a_{v''}) \\ & + w_v \sum_{v' \in N(v)} \log p(v > v') + \sum_{v' \in N(v)} w_{v'} \log p(v' > v) + const \end{aligned} \tag{5.8}$$

where $a_{v''}$ is the area associated with neighbor v'' of venue v. $N(v) \setminus a_v$ is the set of neighbors of venue v but not in area a_v. We explain each component in Equation 5.8 as follows:

- The first component $\sum_{\substack{i \in U, \\ v' \in a_v}} w_{iv'} \log p(i \to a_v)$ indicates the number of times user i checks into venues in area a_v (including venue v).
- The second component $\sum_{\substack{i \in U, \\ v'' \in N(v) \setminus a_v}} w_{iv''} \log p(i \to a_{v''})$ represents the number of times user i checks into venues in the adjacent areas of area a_v.
- The third component $w_v \sum_{v' \in N(v)} \log p(v > v')$ models the winning of venue v over its neighbors.
- The fourth component $\sum_{v' \in N(v)} w_{v'} \log p(v' > v)$ models the losing of venue v to its neighbors.
- Finally, *const* is the constant which is independent of σ_v and it will disappear after taking derivative of log-likelihood with respect to σ_v.

There is no closed-form solution for the optimization problem in Equation 5.7. We therefore use *gradient descent* to find the local optimal solution. Consequently, the derivative of log likelihood with respect to σ_v is:

$$\begin{aligned}\frac{\partial}{\partial\sigma_v}\mathcal{L}(\sigma_v) = \\ \sum_{\substack{i\in U,\\ v'\in a_v}} w_{iv'}\frac{\partial \log p(i\to a_v)}{\partial\sigma_v} + \sum_{\substack{i\in U,\\ v''\in N(v)\setminus a_v}} w_{iv''}\frac{\partial \log p(i\to a_{v''})}{\partial\sigma_v} \\ + w_v\sum_{v'\in N(v)}\frac{\partial \log p(v>v')}{\partial\sigma_v} + \sum_{v'\in N(v)} w_{v'}\frac{\partial \log p(v'>v)}{\partial\sigma_v}\end{aligned} \tag{5.9}$$

Before showing the derivative of each component in log likelihood, we show the derivatives of CDF and Sigmoid functions.

$$\frac{\partial}{\partial\sigma_v}\log\int_{-\infty}^{\sigma_v-\sigma_n}\mathcal{N}(x;0,1)\,\mathrm{d}x = \frac{\mathcal{N}(\sigma_v-\sigma_n;0,1)}{\int_{-\infty}^{\sigma_v-\sigma_n}\mathcal{N}(x;0,1)\,\mathrm{d}x} \tag{5.10}$$

$$\frac{\partial}{\partial\sigma_v}\log S(\sigma_v-\sigma_n) = 1 - S(\sigma_v-\sigma_n) \tag{5.11}$$

We denote $d^2(i,a_v) = (x_i - x_{a_v})^2 + (y_i - y_{a_v})^2$ and $d^2(i,a_{v''}) = (x_i - x_{a_{v''}})^2 + (y_i - y_{a_{v''}})^2$ and the derivatives of two first components of $\mathcal{L}(\sigma_v)$ are

$$\begin{aligned}\frac{\partial}{\partial\sigma_v}\log p(i\to a_v) &= -\frac{2}{\sigma_{a_v}}\frac{\partial\sigma_{a_v}}{\partial\sigma_v} + \frac{1}{\sigma_{a_v}^3}\frac{\partial\sigma_{a_v}}{\partial\sigma_v}d^2(i,a_v) \\ &= -\frac{2}{\sigma_{a_v}^2}\sigma_v + \frac{\sigma_v}{\sigma_{a_v}^4}d^2(i,a_v) \\ \frac{\partial}{\partial\sigma_v}\log p(i\to a_{v''}) &= -\frac{2}{\sigma_{a_{v''}}}\frac{\partial\sigma_{a_{v''}}}{\partial\sigma_v} + \frac{1}{\sigma_{a_{v''}}^3}\frac{\partial\sigma_{a_{v''}}}{\partial\sigma_v}d^2(i,a_{v''}) \\ &= -\frac{2}{\sigma_{a_{v''}}^2}\sigma_v + \frac{\sigma_v}{\sigma_{a_{v''}}^4}d^2(i,a_{v''})\end{aligned} \tag{5.12}$$

In the case of VAN_{CDF}, from Equation 5.10, we have

$$\begin{aligned}\frac{\partial}{\partial\sigma_v}\log p(v>v') &= \frac{\mathcal{N}(\sigma_v-\sigma_{v'};0,1)}{\int_{\infty}^{\sigma_v-\sigma_{v'}}\mathcal{N}(x;0,1)\,\mathrm{d}x} \\ \frac{\partial}{\partial\sigma_v}\log p(v'>v) &= -\frac{\mathcal{N}(\sigma_{v'}-\sigma_v;0,1)}{\int_{\infty}^{\sigma_{v'}-\sigma_v}\mathcal{N}(x;0,1)\,\mathrm{d}x}\end{aligned} \tag{5.13}$$

In the case of $VAN_{Sigmoid}$, from Equation 5.11, we have

$$\begin{aligned}\frac{\partial}{\partial\sigma_v}\log p(v>v') &= 1 - S(\sigma_v-\sigma_{v'}) \\ \frac{\partial}{\partial\sigma_v}\log p(v'>v) &= -\left(1 - S(\sigma_{v'}-\sigma_v)\right)\end{aligned} \tag{5.14}$$

During *gradient descent*, we also use *back-tracking* technique [10] to find the best *learning rate* to fit our model.

5.3 Implementation Note

There are some implementation tricks for efficiently updating user locations and competitiveness of venues in parameter learning.

- *Update of user locations:* Equations 5.5 and 5.6 update latitude and longitude of user home locations respectively. These equations derive home locations of a user from the center and attractiveness of areas where the user performs check-ins, and the number of check-ins of the user to the areas. We do not need information from the other users at all. Hence, we could update the home locations of different users simultaneously.

- *Update of competitiveness of venues:* Since the number of venues in the dataset is always large, the parameter learning of our model may incur much running time. For this reason, to infer the competitiveness σ_v of venue v, we assume that the competitiveness of other venues in V are constant. Gradient descent will then be applied to search for the optimal σ_v. In this way, we could parallelize the update of competitiveness of all venues.

Algorithm 1 summarizes the parameter learning of **VAN** model. We split users in U into two subsets U_k and U_n, i.e. $U = U_k \bigcup U_n$. U_k and U_n denote the subsets of users whose home locations are known and unknown respectively.

Convergence Analysis: Suppose we have an initial value of log likelihood. After updating location of users in U_n by Equations 5.5 and 5.6 (from steps 5 to 7), the log likelihood will increase because it moves along the gradient direction of x_i and y_i $\forall i \in U_n$. Gradient descent with backtracking updates the competitiveness of σ_v (from steps 8 to 10). Thus, the log likelihood will always converge to the stationary point.

input : set of users U_k and their locations; set of venues V and their locations; set of check-ins C; area size s
output: $\{\sigma_v\}_{v \in V}$; $\{(x_i, y_i)\}_{i \in U_n}$

for $v \in V$ **do**
  `// initialize to positive value`
  $\sigma_v = const$;
end
while *Log-likelihood is not convergent* **do**
  for $i \in U_n$ **do in parallel**
    Update x_i and y_i by Equation 5.5 and Equation 5.6;
  endfor
  for $v \in V$ **do in parallel**
    Update σ_v by *gradient descent* with *back-tracking*;
  endfor
  Calculate Log-likelihood by Equation 5.4;
end

Algorithm 1: Parameter Learning of **VAN** model

5.4 Evaluation using Synthetic Data

In this section, we will create a synthetic dataset to evaluate the performance of the VAN models in: (i) recovering of venue competitiveness, and (ii) prediction of user home locations. Moreover, the **neighbors** of a venue v, $N(v)$, are venues which are within area a_v and the areas adjacent to a_v denoted by $adj(a_v)$. That is, $N(v) = \{v' | v' \in adj(a_v)\} \cup \{v' | v' \in a_v\} \setminus \{v\}$. We consider the venues in $adj(a_v)$ as neighbors because we want to include venues in these nearby area as competitors of v.

5.4.1 Data Generation

The synthetic dataset is loosely constructed using the principles the VAN models are based on. We want to evaluate the robustness of the models and also their accuracy in prediction tasks.

Based on the map of Singapore, we generate venues, users and check-ins using a set of parameters listed in Table 5.2. To keep the dataset simple, every user has the same number of check-ins n_c which is one of the data generation parameters.

Table 5.2: Model Parameters of synthetic data.

Model Parameters	Symbol
Number of users	n_i
Number of venues	n_v
Number of check-in per user	n_c
Size of area	s
Variance	ρ

The data generation process follows the steps below.

- *Venues:* We randomly select a location for each venue. For each venue v, we generate its competitiveness σ_v following a Gaussian distribution with mean at the center of the map and variance ρ. The larger the value of ρ, the more concentration are the competitive venues at the center of the map.

- *Users:* For each user i, we randomly select one venue as his/her home location.

- For each pair of user i and venue v, we derive a *pseudo-probability* p_{iv} in two steps. Firstly, i selects the area a_v with probability $\frac{\sum_{v' \epsilon a_v} \sigma_{v'}}{d_{i,a_v}}$ where d_{i,a_v} is the distance between user i and area a_v. Secondly, among venues in area a_v, user i chooses venue v with probability $\frac{\sigma_v}{\sum_{v' \epsilon a_v} \sigma_{v'}}$. Formally, $p_{iv} = \frac{\sum_{v' \epsilon a_v} \sigma_{v'}}{d_{i,a_v}} \frac{\sigma_v}{\sum_{v' \epsilon a_v} \sigma_{v'}} = \frac{\sigma_v}{d_{i,a_v}}$. The intuition behind is to create the effects of distance, area attractiveness and neighborhood competitiveness.

- Finally, the number of check-in n_{iv} between user i and venue v is generated by $n_{iv} = n_c \cdot \frac{p_{iv}}{\sum_{v'} p_{iv'}}$. Moreover, n_{iv} is rounded down if it is not an integer.

5.4.2 Evaluation

Among the data generation parameters, we empirically fix $n_i = 50$, $n_v = 500$, $n_c = 2000$ and $s = 0.1$ geography degree (around 10 kilometers). We vary the ρ parameter in our experiments below:

- **Competitiveness Prediction:** In the first experiment, we hide the competitiveness of all venues. We apply the $\mathbf{VAN}_{Sigmoid}$ and $\mathbf{VAN}_{CDF}$ models on the known users' home locations, venue locations and check-ins to recover the venues' competitiveness.

  We evaluate the accuracy of results by **Pearson correlation** between the actual venue competitiveness and the learnt ones. We do not evaluate using the actual competitiveness values as they depend on the initial value assignment. As we evaluate the competitiveness ranking, we introduce a baseline model *CCount* which ranks the venues by the number of check-ins received from the users.

  As shown in Table 5.3, the two variants of our model always outperform the baseline. Moreover, as we increase ρ, the performances of all models drop. The reason is that larger ρ sees check-ins distributed equally among the areas. This distribution is harder for any model to infer the competitiveness ranking correctly. Between $\mathbf{VAN}_{Sigmoid}$ and $\mathbf{VAN}_{CDF}$ models, there is however no clear winner.

- **Home Location Prediction:** In the second experiment, we evaluate the $\mathbf{VAN}_{Sigmoid}$ and $\mathbf{VAN}_{CDF}$ models in home location prediction task. In this task, we hide all home locations of users and use the models to recover them. We use a few simple methods, namely, *center of the mass (COM)* and *most check-in venue (MCV)* as baselines. We do not use more complicated techniques [14] as these techniques require more input parameters (i.e. time of check-ins) and are less general to compare with our model.

  The user home locations are updated by Equations 5.5 and 5.6 regardless of Sigmoid or CDF function. The reason is that in this experiment, the attractiveness of each area can be inferred by the competitiveness of the venues inside, the number of check-ins and location of areas from the

dataset.

The last three columns of Table 5.3 show the home location prediction accuracy of our models and baselines. We measure the **error distance** defined by the average distance from predicted home location to actual home locations of users. From the result in Table 5.3, we conclude that our model outperforms the two baselines, making 57.7% and 28.8% improvement over the *COM* and the *MCV* methods respectively.

Table 5.3: Result of synthetic data with different ρ of VAN model. The best result is highlighted.

	Pearson correlation			Error distance(km)		
ρ	$VAN_{Sigmoid}$	VAN_{CDF}	$CCount$	VAN	COM	MCV
0.1	0.85	**0.87**	0.71	**5.2**	8.2	6.7
0.5	**0.66**	0.65	0.63	**5.4**	7.4	6.8
1.0	**0.44**	0.33	0.28	**6.1**	8.1	7.2

From the experiment results of VAN model in the above two tasks, we conclude that VAN can recover the competitiveness of venues as well as the users' home locations. Moreover, VAN also achieves good result even if the data generation process does not follow strictly to the model.

5.5 Evaluation using Real Data

We evaluate our proposed VAN models on real datasets in four separate tasks. We first conduct experiments to evaluate the VAN models in home location prediction task. We also evaluate the venue competitiveness learned by VAN models using some case studies. Next, we conduct another experiment to evaluate the VAN models in check-in prediction task. Lastly, we show the robustness of models when area boundaries are modified.

Similar to the experiments in Section 5.4, the geography degree is chosen as the unit of parameter s. Moreover, the definition of the **neighbors** of a venue v, $N(v)$, is adopted from Section 5.4.

5.5.1 Home Location Prediction

Description: In this task, we aim to predict the home locations of users using our VAN models and some baselines. Among the baseline methods for comparison the is **PMM** model is the state-of-the-art home location prediction method.

Setup: In total, we have the exact home locations of 856 users in **H_SG** dataset. However, there are 341 of them whose home locations cannot be predicted by PMM model as these users have too few check-ins or too few venues not giving PMM enough data to learn their home locations. Hence, we will conduct the experiment on the remaining 515 users.

In the experiment, we separate 515 users into five folds each with 103 users. For each run, we hide the home location of users in one fold and use all check-in data from all five folds and home location of users from the remaining four folds as input. Each model will then predict the home location of users in the hidden fold.

For PMM, only the check-ins of users are used to predict their home locations. Hence, each time, we select one fold and predict home locations of users in that fold by their check-in data.

Similar to **H_SG**, there are 154 out of 455 users in **H_JK** whose home locations could be predicted by PMM. We therefore divide them into five folds in the experiment.

Note that our model could perform over the entire dataset but to guarantee fairness, we only conduct this experiment over the subset of users in which PMM could predict in both datasets.

Baselines: We consider several baselines below in this home location prediction task.

- **Center of the mass** (**COM**): This model returns the center of the mass of all check-ins of a user as his/her home location. Formally, a user with n check-ins at (x_i, y_i)'s has home location predicted at $x_c = \frac{\sum_{i=1}^{n} x_i}{n}$ and

$y_c = \frac{\sum_{i=1}^{n} y_i}{n}$.

- **Most check-in venue** (**MCV**): This model selects the most frequent check-in venue of a user as his/her home location.

- **Periodic Mixture Model** (**PMM**): This model was proposed by Cho et al. [14] and it groups check-ins of a user into two clusters named *home* and *work*. The *Home* cluster represents non-working hours check-ins. We return the center of home cluster as the predicted home location of the user.

Performance Measure: We measure the *distance* between the predicted home venue p_i and the actual home location h_i of user i. The overall performance is thus defined by the **average error distance** (**AED**) between all predicted home locations and actual home locations. Moreover, we define another metric $prec@k$ is ratio of users whose distance from their predicted location to actual home is less than k.

$$error_m = \frac{\sum_{i \epsilon U} dist(p_i, h_i)}{|U|}; prec@k = \frac{|\{i : dist(p_i, h_i) < k\}|}{|U|} \tag{5.15}$$

where $dist(\cdot, \cdot)$ returns the physical distance between two locations by haversine formula. In our experiment, we choose $k = 5km$.

Result: Table 5.4 depicts the performance of baselines and our models with different s parameter values in the two datasets **H_SG** and **H_JK**.

In the case of **H_SG** dataset, our $\mathbf{VAN}_{Sigmoid}$ and $\mathbf{VAN}_{CDF}$ models outperform **PMM** model by 12.34% and 13.16%, respectively. Compared with other baselines, the VAN models yield accuracy with up to 28% improvement. The superior performance of VAN models is not affected by the s parameter.

For **H_JK**, we observe that the performance of our VAN models is affected by the s parameter setting. The optimal s value is 0.025. Under this setting, our VAN models outperform PMM and other baselines. The reason for the

poorer performance in this dataset may be due to the sparsity of check-ins in this dataset.

The poor performance of COM and MCV in both datasets compared to VAN model comfirms the result of synthetic data. However, we do not have the groundtruth of competitiveness of venues so we cannot compare like synthetic data.

Table 5.4: Home prediction result of **H_SG** and **H_JK**. Metric of error in this table is meter. *prec*@5km is surrounded by brackets. The best result of each dataset is highlighted.

		AED(prec@5km)	
	s	**H_SG**	**H_JK**
COM	-	6570.3 (46.2%)	5564.4(43.4%)
MCV	-	7117.7 (40.3%)	5547.2 (45.5%)
PMM	-	6126.3(49.3%)	4823.2(60.8%)
$VAN_{Sigmoid}$	0.1	5561.8(50.7%)	5623.8(53.3%)
	0.05	**5046.4**(59.8%)	5125.2(60.4%)
	0.025	5475.2(56.7%)	4757.8(64.4%)
VAN_{CDF}	0.1	5564.6(51.46%)	5331.1(56.1%)
	0.05	5181.6(**60.4%**)	4866.1 (59.1%)
	0.025	5213.8(56.9%)	**4357.2**(**68.2%**)

5.5.2 Venue Competitiveness Ranking

In this section, we will use the check-in data of 856 users including their home locations and the locations of venues to infer the competitiveness of all venues in **H_SG** dataset. The venues ranking is ordered by decreasing competitiveness values.

Table 5.5 shows the top 15 venues based on the competitiveness values learned by $\mathbf{VAN}_{CDF}$ model with $s = 0.1$. Due to the lack of language knowledge, we only conduct the evaluation with this configuration for **H_SG** and the result of **H_JK** is not included.

Queenstown MRT Station receives 241 check-ins from six users but the check-ins are not evenly distributed among them. Most of check-ins are from one user. He is an active user who has 770 check-ins on 112 venues but 231 of

Table 5.5: Top 15 venues of $\mathbf{VAN}_{CDF}$ for **H_SG** with s = 0.1. The third column is the competitiveness value of venues.

Rank	Venue Name	σ_v	# of users	# of check-ins	check-ins per user
1	Nex Serangoon	6.36	128	476	3.71
2	Cineleisure Orchard	6.28	148	397	2.68
3	ITE College Central	6.23	41	361	8.80
4	VivoCity	6.19	158	296	1.87
5	Ngee Ann Polytechnic	6.14	22	293	13.3
6	Bugis Junction	6.097	129	242	1.87
7	ION Orchard	6.096	135	271	2
8	Queenstown MRT Station	6.095	6	241	40.1
9	E!hub Downtown East	6.09	53	275	5.18
10	Singapore Changi Airport	6.06	139	258	1.85
11	Plaza Singapura	6.05	114	249	2.18
12	AMK Hub	6.04	92	242	2.63
13	Jurong Point	6.03	88	235	2.67
14	313@somerset	6.026	114	235	2.06
15	Causeway Point	6.026	80	240	3

his check-ins are on *Queenstown MRT Station*. Moreover, his home location is also near to *Queenstown MRT Station*, i.e., 210 meters away. This is the outlier case in our dataset. This case example shows a weakness of our model which cannot handle the case of significant number of check-ins from one or very few users.

To understand the high competitiveness rank of *Ngee Ann Polytechnic* and *ITE College Central*, we take a look at their users who made check-ins to these places. The result is not surprising. Most users of these places are young people and students of these schools. Moreover, these people are living around Singapore. Therefore, these schools gain more competitiveness from the students who perform frequent check-ins.

For other cases in Table 5.5, they are crowded hubs of Singapore where tourists and local people visit. For example, *Singapore Changi Airport* is an international airport which sees more than 54 millions passenger movements per year. All the others venues are popular shopping malls in Singapore.

To quantify the ranking performance, we compare our ranking with

Table 5.6: The correlation of Foursquare score and $\mathbf{VAN}_{CDF}$ model and *Cks Model* through Jaccard similarity score. The best performance is highlighted.

Metric	VAN	*Cks Model*
$Jaccard$@20	**5.21%**	3.15%
$Jaccard$@50	**6.41%**	4.64%
$Jaccard$@100	**8.89%**	6.82%

Foursquare score which is the aggregate score from the feedbacks of users. The baseline is number of check-ins denoted as *Cks Model*. In *Cks Model*, the more check-ins the venue has, the higher rank it is. The metric is $Jaccard@k$. Particularly, we select the top-k venues by each ranking and compute the Jaccard score with top-k venues returned by Foursquare scores. The higher the value, the better the model. Table 5.6 shows the performance of VAN model and *Cks Model*. Across different values of k, the performance of VAN model is better than *Cks Model* since it is closer to the ranking of the feedback of users represented by Foursquare score.

5.5.3 Check-in Prediction Task

In this part, we will present the result of check-in prediction task. This task predicts check-ins between users and venues.

Setup: We sort check-ins in the **H_SG** and **H_JK** datasets by time and then divide each dataset into 10 folds. For each iteration, we hide one fold as test set and use the remaining nine folds as training set.

Baselines: In order to compare the performance of our model, we use some baselines below

- Probabilistic Matrix Factorization *PMF*: It was proposed by Mnih *et. al.* [62] and was widely adopted to many research areas including check-in prediction. Its idea is to factorize check-in matrix of users and venues into user-feature and venue-feature matrix alone. The parameter for this method is K the number of features for user-matrix and venue-matrix so we use the default value K = 10.

- Multi-center Gaussian Model *MGM*: Cheng *et. al.* [12] proposed a check-in prediction method based on multiple Gaussian distributions. Its main idea is to construct the centers of activity of users and each center is represented by a Gaussian distribution. Thus, its idea of areas is similar to our model but in our case, we pre-define areas for all venues while *MGM* automatically detects areas for each user. To detect the clusters for *MGM*, we apply the non-parametric method from Blei *et. al.* [7] which brings us fast speed via variational inference and the number of clusters automatically. For α parameter of *MGM*, we use the default value $\alpha = 0.2$ to run experiment.

- Fusion Framework *PMF-MGM*: It is the combination between matrix factorization and *MGM* [12]. Cheng *et. al.* reported that fusion framework outperforms *PMF* and *MGM* models. Thus, we use *PMF* with *MGM* as its component to predict the check-in of users.

- Matrix Factorization with Neighborhood Influence *N-MF*: Hu. *et. al.* [35] studied the intrinsic and extrinsic characteristics of geographical neighbors upon the matrix factorization framework. We use the default number of latent features $K = 20$ and two venues are neighbor if their distance is less then a predefined threshold d. In our experiment, we examine 100 meter and 200 meter as the value of d.

- Exposure Matrix Factorization with locations as exposure covariates *Expo-MF*: The model of Liang *et. al.* [57] is the state-of-the-art variance of matrix factorization to investigate user exposure[1]. It can incorporate the location of venues in order to increase the performance. Similar to their experiment, we apply K-Means to cluster venues, the location vector of each venue is its probability to each cluster. We use the default number of latent features $K = 100$ and it is also equal to the number of

[1]https://github.com/dawenl/expo-mf

clusters in K-Means.

Performance Measure: After training, for each user, we select the top k venues predicted by each method and compare against all the venues checked in by the users in the test data. Note that the user may have more or less than k check in venues in the test data. We use $recall@k$ and $precision@k$ as the metric to compare the performance of our model and the baselines. Finally, we report the average values of each metric for all folds.

$$\begin{aligned} recall@k &= \frac{1}{|U|}\sum_{u\in U}\frac{|\mathcal{L}(u,k)\cap\mathcal{L}^t(u)|}{|\mathcal{L}^t(u)|} \\ precision@k &= \frac{1}{|U|}\sum_{u\in U}\frac{|\mathcal{L}(u,k)\cap\mathcal{L}^t(u)|}{k} \end{aligned} \tag{5.16}$$

where $\mathcal{L}(u,k)$ is the top k venues of each user u of each predictive method; $\mathcal{L}^t(u)$ represents set of venues of user u in test set and $|\cdot|$ returns the number of elements of set.

Results: The result of check-in prediction task for two datasets **H_SG** and **H_JK** are shown in Table 5.7 . In our experiment, our model with Sigmoid or CDF function always outperforms all baselines in both datasets. For instance, in **H_SG**, our model could reach up to three times better than PMF and 10 times better than MGM. Overall, in both datasets, if we reduce the size of area, the performance of VAN model decreases. Specifically, the performance of size of 0.05 is usually better than the one of size of 0.025 but less accuracy than size of 0.1. Additionally, the result of VAN_{CDF} is usually better than the performance of $VAN_{Sigmoid}$. Between two baselines, MGM has better performance than PMF in **H_JK** dataset but in **H_SG**, the result of MGM does not overcome the one of PMF. PMF-MGM is the hybrid of MGM and PMF so its performance is in the middle of both models.

5.5.4 Area Boundary Shift

In this section, we examine the robustness of our model by shifting the area without changing the area size. Specifically, we shift the area and measure the check-in prediction performance of our model.

Setup: Since we evaluate by check-in prediction task, we reuse the setup and performance metrics (i.e. *recall*@k and *precision*@k) from the previous tasks. Recall that we create areas by dividing the city into grid cells of equal width. The boundaries of areas are defined by the vertical and horizontal lines sharing the same longitudes and latitudes, respectively. As the choice of these boundary lines can change, we would like to know if shifting the grid cells could affect the check-in prediction performance of our model. We choose VAN_{CDF} to be examined in this experiment since it has the best check-in performance (see Table 5.7). We use VAN_x and VAN_y to denote our model if grid cells shift 0.05 degree on latitude and longitude respectively. Finally, we denote VAN_{xy} is the model if the shifting is 0.05 on both latitude and longitude simultaneously. Since the move is one half of the area width, a shift in either direction leads to the same form of grid cell generation.

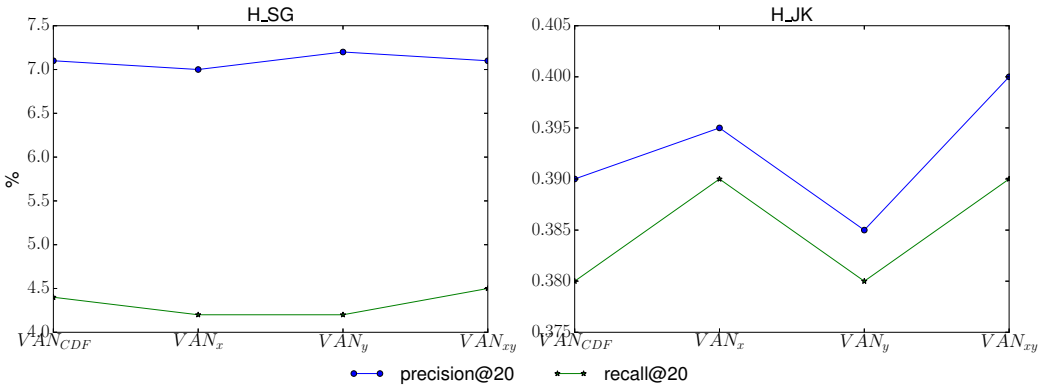


Figure 5.2: *precision*@20 and *recall*@20 in **H_SG** and **H_JK** of VAN_{CDF} with s = 0.1 under different ways of constructing areas.

Result: Figure 5.2 shows the performance of VAN_{CDF} with s = 0.1 under different ways of constructing areas. From the figure, we firstly observe that despite of shifting the grid, the performance of VAN is stable under pre-

cision and recall metric since the difference of our original construction and the shifting ones are under 5%. Specifically, the maximum difference between VAN_{CDF} and its shifting variant models is 1.42% under *precision*@20 metric and 4.55% under *recall*@20 metric in **H_SG** dataset. Secondly, in both datasets, the performance differences among various models are less than 5%. From these observations above, we can conclude that VAN model is robust under area shifting.

Table 5.7: The performance (*precision@k* and *recall@k*) of **H_SG** and **H_JK** datasets in check-in prediction task. The *recall@k* values are put between brackets. We highlight the best result for each value of *k*.

		precision@k(*recall@k*)										
		VAN_{CDF}			$VAN_{Sigmoid}$			PMF	MGM	PMF-MGM	N-MF		Expo-MF
	k / s	0.1	0.05	0.025	0.1	0.05	0.025				100m	200m	
H_SG	20	**7.1%** (**4.4%**)	3.1% (1.9%)	1.63% (1.1%)	6.6% (4.2 %)	3.5% (1.95%)	0.7% (0.48%)	2.14% (1.4%)	0.32% (0.36%)	2.05% (1.35%)	0.8% (0.3%)	0.75% (0.29%)	2.7% (1.5%)
	50	**5.6%** (**8.7%**)	2.8% (4.2%)	1.89% (3.1%)	**5.6%** (8.67%)	3% (4.62%)	1.1% (1.67%)	1.52% (2.6%)	0.2% (0.61%)	1.48% (2.5 %)	0.68% (0.8%)	0.6% (0.75%)	2.1% (1.6%)
	100	3.9% (12.1%)	2.23% (6.6%)	1.88% (6 %)	**4%** (**12.26%**)	3.5% (10.53%)	1.7% (5.65%)	1.1% (3.8%)	0.19% (1.11%)	1.08% (3.7 %)	0.57% (1.4%)	0.54% (1.35%)	1.2% (2.3%)
H_JK	20	0.39% (0.38%)	0.24% (0.56%)	0.27% (0.8%)	0.35% (0.36%)	0.26% (0.74%)	0.29% (**0.96%**)	0.29% (0.14%)	0.26% (0.24%)	0.29% (0.14%)	**0.7%** (0.57%)	**0.8%** (0.7%)	0.2% (0.3%)
	50	**0.72%** (**2.2%**)	0.26% (1.9 %)	0.28% (1.6%)	0.58% (1.73%)	0.23% (1.74%)	0.24% (1.51%)	0.28% (0.29%)	0.3% (0.7 %)	0.28% (0.3 %)	0.4% (0.96%)	0.4% (1.2%)	0.22% (1.1%)
	100	**0.63%** (**4.3%**)	0.43% (4 %)	0.28% (3.2%)	0.58% (3.8%)	0.38% (3.49%)	0.23% (3 %)	0.28% (1.12%)	0.34% (1.8 %)	0.28% (1.1 %)	0.29% (1.4%)	0.3% (1.7%)	0.2% (2.5%)

Chapter 6

Modeling Neighborhood Competition and Area Attraction with Latent Features

So far, we have modeled *neighborhood competition* and *area attraction* using a Bayesian approach. The VAN models developed in this approach have not considered user preference and venue topics which are latent. As VAN models also assume the availability of user home location information, they cannot be applied in application scenarios that do not have such information. In this chapter, we therefore develop an improved model by (1) discarding the user home location assumption and dropping distance effect from model design; and (2) incorporating the user and venue latent factors to enhance the modeling of neighborhood competition.

6.1 Proposed Model

In this section, we propose a model called ***V****isitation by* ***A****ttractiveness and* ***N****eighborhood competition* **F**actorization (VANF). The VANF model is an extension of standard non-negative matrix factorization to model check-in behavior incorporating area attraction, and neighborhood competition. The VANF

Table 6.1: Table of Notations.

Notations	**Meaning**
U, V, C	set of all users, venues and check-ins
U_i	latent feature vector of user i
V_v	latent feature vector of venue v
w_{iv}	number of check-in of user i to venue v
w_v	total number of check-in of venue v
a_v	area a_v containing venue v
s	the width of area
$N(v)$	set of neighbor venues of v
$L_a(\cdot)$	Logistic function with steepness a
$p(i \rightarrow a_v)$	probability of user i visiting area a_v
λ_u, λ_v, λ_f	regularization of user, venue vectors and friendship

also incorporates social homophily effect when users are connected with one another. In Section 6.1.1, we will first define the important concepts in the VANF model and its model assumptions. We then introduce the model formally in Section 6.1.2. The learning of VANF model parameters is given in Section 6.1.3.

6.1.1 Model Description

In the VANF model, we model each user i or venue v as a vector of latent features U_i and V_v respectively. When user i and venue v have preferences on similar latent features, $U_i^T V_v$ returns a large value implying that user i is likely to perform check-in on venue v. We also use w_{iv} to denote the number of check-ins by user i on venue v. Readers can refer to Table 6.1 for the notations used in the VANF model.

To model area attraction, we again divide the city into mutually exclusive square grid cells of width s. We use a_v to denote the square or *area* which contains v. The **VANF** model makes the following assumptions for each check-in between a user and a venue:

- First of all, every user chooses an area to perform a check-in based on a combination of area attractiveness and the user's preference on the area. Area attractiveness is a quantitative measure defined to capture how well

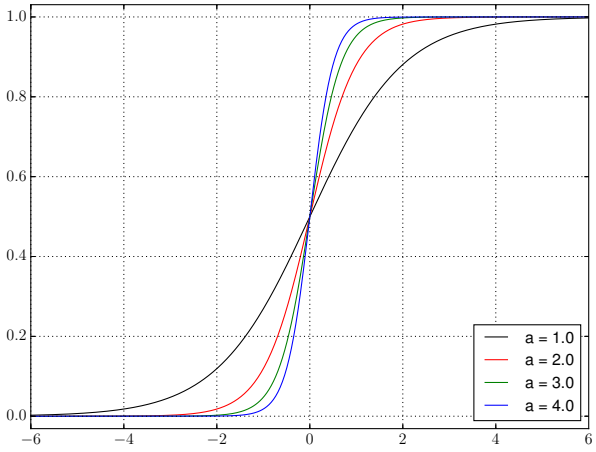


Figure 6.1: Logistic function $f(x) = \dfrac{1}{1+\exp(-a \cdot x)}$ with different values of steepness a.

the area can attract users based on the venues within the area.

- Secondly, every venue inside an area must compete against its neighboring venues in order to gain a check-in from the user.

The **neighbors** of a venue v, denoted as $N(v)$, are venues within a_v and the areas adjacent to a_v are denoted by $Adj(a_v)$. That is, $N(v) = \{v' | v' \in Adj(a_v)\} \cup \{v' | v' \in a_v\} \setminus \{v\}$. We consider the venues in $Adj(a_v)$ as neighbors because we want to include venues in these nearby areas as competitors of v even when v is near the border of a_v.

For a user i, the **attractiveness** $\sigma^i_{a_v}$ of area a_v is defined by the summation of the interaction between the user preference U_i and each latent features $V_{v'}$ of venue v' inside an area a_v. That is, $\sigma^i_{a_v} = \sum_{v' \in a_v} U_i^T V_{v'}$. It means that the venues inside the area collectively attract a check-in from user i.

Every check-in of user i to venue v follows a two-step process. Firstly, user i must select the area a_v. Secondly, the venue v in area a_v must win over all other neighboring venues in $N(v)$ to gain a check-in from user i.

- User i selects the area a_v under the effect of attractiveness $\sigma^i_{a_v}$ of area a_v. We represent this by assigning a probability which is proportion to $\sigma^i_{a_v}$.

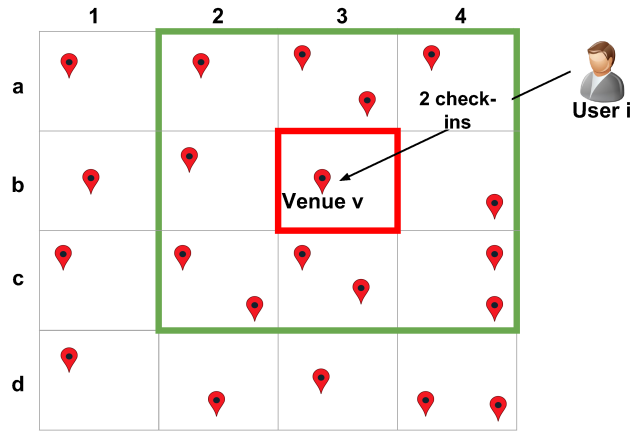


Figure 6.2: Example of Check-in graph.

- To model the winning of venue v over its neighbors, we need to employ the preference of user i to determine if the check-in is performed or not. We assume that if the latent similarity between user i and venue v is higher than the one between user i and the neighbors v' of v, the probability that i visits v (denoted as $p_i(v > v')$) is higher than the one between i and v' . We therefore map the value of $U_i^T V_v - U_i^T V_{v'}$ to interval [0, 1] so as to model $p_i(v > v')$. When $p_i(v > v') > p_i(v' > v)$, user i is likely to make check-in on v rather than v'. We define $p_i(v > v') = L_a(U_i^T V_v - U_i^T V_{v'}) = \frac{1}{1 + \exp(-a(U_i^T V_v - U_i^T V_{v'}))}$ where L_a is a logistic function [38] with steepness parameter a. Logistic function is a function family which Sigmoid function belongs to. Sigmoid function is a logistic function with $a = 1$. When a goes to infinity, logistic function turns into an indicator function. Figure 6.1 shows logistic function with different values of steepness.

Example: Figure 6.2 depicts two check-ins at venue v by user i, i.e. $w_{iv} = 2$. To perform each check-in at venue v, user i has to select area $(b, 3)$ (enclosed by a red box) considering the similarity between the preference of user i and the venues within the area. Moreover, venue v needs to *win* over all of its neighbors in the adjacent areas enclosed by the square box in green.

6.1.2 Model Formalization

We now formally define the VANF model. In the VANF model, the probability p_{iv} of a check-in from user i to venue v is defined by the following formula:

$$p_{iv} = p(i \to a_v) \prod_{v'' \in N(v)} p_i(v > v'') \tag{6.1}$$

Equation 6.1 says that p_{iv} consists of two components: $p(i \to a_v)$ denoting the probability of user i selecting area a_v and $p_i(v > v'')$ representing the probability that user i prefers to perform check-in on venue v over its neighbor v''.

Recall that U_i and V_v denote the latent feature vectors of user i and venue v respectively. We thus define $p(i \to a_v)$ as

$$p(i \to a_v) = \sum_{v' \in a_v} p(v'|i) = \sigma^i_{a_v} = \sum_{v' \in a_v} U_i^T V_{v'} \tag{6.2}$$

The second component of Equation 6.1 is defined as:

$$p_i(v > v'') = L_a(U_i^T V_v - U_i^T V_{v''}) \tag{6.3}$$

By substituting the components in Equation 6.1, we have:

$$\begin{aligned} p_{iv} &= \left(\sum_{v' \in a_v} p(v'|i) \right) \prod_{v'' \in N_v} p_i(v > v'') \\ &= \left(\sum_{v' \in a_v} U_i^T V_{v'} \right) \prod_{v'' \in N_v} L_a(U_i^T V_v - U_i^T V_{v''}) \\ \log p_{iv} &= \log \sum_{v' \in a_v} U_i^T V_{v'} + \sum_{v'' \in N_v} \log L_a(U_i^T V_v - U_i^T V_{v''}) \end{aligned} \tag{6.4}$$

Next, we define the log-likelihood $\mathcal{L}(C)$ of a set of check-ins C from users of U on venues of V has the following form:

$$\mathcal{L}(C) = \sum_{(i,v) \in C} w_{iv} \log p_{iv} = L_1(C) + L_2(C) \tag{6.5}$$

where

$$
\begin{aligned}
L_1(C) &= \sum_{(i,v)\in C} w_{iv} \log\Big(\sum_{v'\in a_v} U_i^T V_{v'}\Big) \\
L_2(C) &= \sum_{(i,v)\in C} w_{iv} \sum_{v''\in N_v} \log L_a(U_i^T V_v - U_i^T V_{v''})
\end{aligned}
\tag{6.6}
$$

To learn the latent features and other variables of users and venues in VANF model, we maximize the log-likelihood defined in Equation 6.5. Formally, we have the optimization problem defined below:

$$
\{U_i^*, V_v^*\}_{i\in U, v\in V} = \arg\max_{i\in U, v\in V} (\mathcal{L}(C) - \lambda(C)) \tag{6.7}
$$

where $\lambda(C)$ is the regularization term that prevents overfitting [26]. In our model, we use L-2 norm for $\lambda(C)$ since it can be solved easily [26] and it is widely applied in matrix factorization method [43, 49, 62]. Formally, $\lambda(C)$ is defined as

$$
\lambda(C) = \lambda_u \sum_i \|U_i\|_2^2 + \lambda_v \sum_v \|V_v\|_2^2 \tag{6.8}
$$

where λ_u and λ_v are the regularization parameters for the latent features of users and venues respectively.

Incorporating Social Homophily: Similar to [12], we model social homophily by adding a social regularizer $\lambda_f \sum_{(i,i')\in F} \|U_i - U_{i'}\|^2$ to Equation 6.7. In other words, if two users i and i' have social connection between them, their latent feature vectors U_i and $U_{i'}$ are expected to be similar. λ_f is the parameter to control the importance of social homophily effect. Formally, we have a new objective function

$$
\{U_i^*, V_v^*\}_{i\in U, v\in V} = \arg\max_{i\in U, v\in V} (\mathcal{L}(C) - \Lambda(C)) \tag{6.9}
$$

where

$$
\Lambda(C) = \lambda(C) + \lambda_f \sum_{(i,i')\in F} \|U_i - U_{i'}\|^2 \tag{6.10}
$$

6.1.3 Model Inference

To solve the optimization problem in Equations 6.7 and 6.9, we use *Stochastic Gradient Descent* (SGD) [10]. SGD is a widely used technique to learn latent features in matrix factorization-based framework [35, 60, 43]

In SGD, we first derive the derivative of the objective function with respect to each variable. Each step of SGD only considers one user-venue pair (i, v).

We firstly select one user-venue pair randomly and take the derivative of user feature vector U_i of the regularization

$$\frac{\partial \Lambda((i,v))}{\partial U_i} = 2\lambda_u U_i + 2\lambda_f \sum_{(i,i')\in F} (U_i - U_{i'}) \tag{6.11}$$

$$\begin{aligned} \frac{\partial L_1((i,v))}{\partial U_i} &= w_{iv} \frac{1}{\sum_{v'\in a_v} U_i^T V_{v'}} \sum_{v'\in a_v} \frac{\partial U_i^T V_{v'}}{\partial U_i} \\ &= w_{iv} \frac{1}{\sum_{v'\in a_v} U_i^T V_{v'}} \sum_{v'\in a_v} V_{v'} \end{aligned} \tag{6.12}$$

$$\frac{\partial L_2((i,v))}{\partial U_i} = w_{iv} \sum_{v''\in N_v} \frac{1}{L_a(U_i^T V_v - U_i^T V_{v''})} \frac{\partial L_a(U_i^T V_v - U_i^T V_{v''})}{\partial U_i} \tag{6.13}$$

To simplify the formula, we introduce $d_{i,v,v''} = U_i^T V_v - U_i^T V_{v''}$. Recall that $L_a(d_{i,v,v''})$ is Logistic function of $d_{i,v,v''}$ with steepness a i.e. $L_a(d_{i,v,v''}) = \frac{1}{1+\exp(-a\ d_{i,v,v''})}$. Hence, we have the derivative of $L_a(d_{i,v,v''})$ respected to U_i:

$$\frac{\partial L_a(d_{i,v,v''})}{\partial U_i} = \frac{a}{(1+\exp(-a\ d_{i,v,v''}))^2} \exp(-a\ d_{i,v,v''})(V_v - V_{v''}) \tag{6.14}$$

Secondly, we take the derivative of V_v. The derivative of regularization is

$$\frac{\partial \Lambda((i,v))}{\partial V_v} = 2\lambda_v V_v \tag{6.15}$$

The derivative of each component of the log-likelihood regarding V_v is

$$\begin{aligned}\frac{\partial L_1(i,v)}{\partial V_v} &= w_{iv}\frac{1}{\sum_{v'\epsilon a_v} U_i^T V_{v'}}U_i + \sum_{v^*\epsilon a_v} w_{iv^*}\frac{1}{\sum_{v'\epsilon a_v} U_i^T V_{v'}}U_i \\ \frac{\partial L_2(i,v)}{\partial V_v} &= w_{iv}\sum_{v''\epsilon N_v}\frac{1}{L_a(d_{i,v,v''})}\frac{\partial L_a(d_{i,v,v''})}{\partial V_v}\end{aligned} \tag{6.16}$$

Therefore, we have the derivative of $L_a(d_{i,v,v''})$ respected to V_v as follow:

$$\frac{\partial L_a(d_{i,v,v''})}{\partial V_v} = \frac{a}{(1+\exp(-a\ d_{i,v,v''}))^2}\exp(-a\ d_{i,v,v''})U_i \tag{6.17}$$

The second step of SGD is to update latent feature vectors of users and venues

$$\begin{aligned}U_i &\leftarrow U_i - \alpha\left(\frac{\partial \mathcal{L}(i,v)}{\partial U_i} - \frac{\partial \Lambda(i,v)}{\partial U_i}\right) \\ V_v &\leftarrow V_v - \alpha\left(\frac{\partial \mathcal{L}(i,v)}{\partial V_v} - \frac{\partial \Lambda(i,v)}{\partial V_v}\right)\end{aligned} \tag{6.18}$$

where α is the learning step parameter of SGD.

Then, we repeat to the first step until the objective function gets convergence.

6.2 Experiments and Results

In the absence of ground truth data, our proposed model VANF will be evaluated via *check-in prediction task* which predicts the number of check-ins for user-venue pairs. We compare the check-in prediction accuracy of our model with other baselines. We will also study the effects of model parameter settings on the model performance. These parameters include the steepness of Logistic function, area width and regularization. The variant of VANF model with social homophily denoted as $VANF_s$ is also evaluated in the next experiment. Finally, we conduct experiment to evaluate the effectiveness of VANF model in venue ranking against the Foursquare venue scores. We also present some latent features of venues learned by VANF.

6.2.1 Experiment Setup

We divide check-in data into training and test sets. We sort check-ins in the **SG**, **JK** and **NYC** datasets by their created time and then divide each dataset into five folds. For each run of experiment, we hide one fold as test set and use the remaining four ones as training set. Particularly, for each run, we use four folds for learning model parameters, and then these learned values are used to predict the number of check-ins between users and venues in the hidden fold.

Performance Measures: We use two sets of metrics to measure the performance of our models as well as the baselines. The first set consists of $recall@k$ and $nDCG@k$ which focus more on top ranked results returned by each model. The second set includes *average precision* (AP) and *area under the curve* (AUC) which measure the overall performance.

After training, for each user, we rank all venues according to their prediction scores returned by each model. The venues visited by the same user in the test data are the ground truth. We then compute the different performance measures based on the predicted venue ranking. The performance measures are averaged over all users. We finally derive the mean of the average performance measures over all the folds. We do not use $precision@k$ because we cannot distinguish between a user disliking a venue and a user not knowing the venue [82].

The formula of $recall@k$ and $nDCG@k$ are presented below:

$$\begin{aligned} recall@k &= \frac{1}{|U|} \sum_{u \in U} \frac{|\mathcal{L}(u,k) \cap \mathcal{L}^{test}(u)|}{|\mathcal{L}^{test}(u)|} \\ nDCG@k &= \frac{1}{|U|} \sum_{u \in U} \frac{DCG@k_u}{IDCG@k_u} \end{aligned} \tag{6.19}$$

where $\mathcal{L}(u,k)$ is the top k venues of each user u returned by the model; $\mathcal{L}^{test}(u)$ represents the set of venues of user u in test set. Function $|\cdot|$ returns the set cardinality.

$$
\begin{aligned}
DCG@k_u &= \sum_{i=1}^{|\mathcal{L}(u,k)|} \frac{2^{rel_{ui}} - 1}{\log_2(i+1)} \\
IDCG@k_u &= \sum_{i=1}^{|\mathcal{L}^{test}(u)|} \frac{2^{rel_{ui}} - 1}{\log_2(i+1)}
\end{aligned}
\tag{6.20}
$$

To measure $DCG@k$, we first select the top k venues of each user returned by each method. rel_{ui} is the relevance score of the i-th rank venue of user u. In our experiment, $rel_{ui} = 1$ if $i \in \mathcal{L}^{test}(u)$; otherwise, $rel_{ui} = 0$. The $nDCG@k_u$ is $DCG@k_u$ normalized by the $DCG@k_u$ of the ideal ranking $IDCG@k_u$ of top-k venues for user u.

The formal definitions of AUC and AP are described below:

$$
\begin{aligned}
AUC &= \frac{1}{|U|} \sum_{u \in U} \frac{1}{|E(u)|} \sum_{(v,v') \in E(u)} \delta(p_{uv} > p_{uv'}) \\
AP &= \frac{1}{|U|} \sum_{u \in U} \sum_{n} (R_n^u - R_{n-1}^u) P_n^u
\end{aligned}
\tag{6.21}
$$

where $E(u) = \{(v, v') | v \in \mathcal{L}^{test}(u) \wedge v' \notin (\mathcal{L}^{test}(u) \cup \mathcal{L}^{train}(u))\}$ and $\mathcal{L}^{train}(u)$ represents the set of venues of user u in training set. In other words, $E(u)$ is the set of venue pairs with the 1st venue element in the test set of user u but the second value element without having any implicit feedbacks from user u. Function $\delta(\cdot)$ is the indicator function that returns 1 if the boolean expression inside is true and 0 otherwise.

AP is average precision metric which summarizes the plot as the weighted mean of precision achieved at each threshold with the increase in recall from the previous threshold used as the weight. In the formula of AP, P_n^u and R_n^u are the precision and recall at the n-th threshold of user u.

Default Parameter Setting: For all experiments, we set the number of latent features to 10. The width of area is $s = 0.01$ geographical degree. The default steepness of Logistic function is $a = 2.0$ since it yields us the best prediction performance for the $VANF$ model (See more details in Sections 6.2.4 and 6.2.6). For regularization, we use the default $\lambda_u = \lambda_v = 0.01$ because it does not bias toward users nor venues. In most of the experiments, we use

Table 6.2: Check-in Prediction Results: We boldface the best results for each performance measure. a = 2.0, s = 0.01, f = 10, $\lambda_u = \lambda_v$ = 0.01 and λ_f = 0.01 for $VANF_s$. The symbol $*$ indicates that $VANF_s$ method performs significantly better than $VANF$ while ‡ indicates $VANF$ or $VANF_s$ performing significantly better than the best baseline.

Metrics	$VANF$	$VANF_s$	PMF	MGM	PMF-MGM	N-MF		Expo-MF	SBPR
						100m	200m		
SG									
recall@20	**7.06%** ‡	**7.71%** *‡	1.93%	1.3%	2.21%	0.93%	0.9%	6.5%	1.17%
recall@50	**10.84%**‡	**11.24%***‡	2.6 %	2.17%	3.12%	1.3%	1.26%	7.8%	1.95%
recall@100	**14.46%**‡	**15.26%***‡	3.42%	3.22%	3.92%	1.61%	1.6%	9.12%	2.4%
nDCG@20	**9.21%** ‡	**9.5%** *‡	5.21%	4.92%	5.08%	1.94%	1.4%	8.69%	3.21%
nDCG@50	**6.9%** ‡	**7.32%** *‡	4.43%	4.05%	4.55%	1.67%	1.07%	6.12%	2.54%
nDCG@100	**6.08%** ‡	**6.85%** *‡	4.13%	3.83%	4.16%	1.11%	0.94%	5.72%	2.03%
AP	**70.21%**‡	**72.11%***‡	61.17%	59.73%	61.81%	54.65%	53.91%	68.11%	53.17%
AUC	**74.18%**‡	**75.05%***‡	60.73%	58.14%	61.9%	55.59%	54.09%	72.08%	51.25%
JK									
recall@20	**3.63%**‡	**4.03%** *‡	2.5%	0.15%	2.8%	0.17%	0.175%	2.7%	0.75%
recall@50	**6.5%**‡	**7.3%** *‡	3.86%	0.23%	3.51%	0.67%	0.8%	4.81%	1.01%
recall@100	**8.75%**‡	**9.87%** *‡	5.81%	0.31%	5.9%	1.8%	1.95%	6.01%	1.78%
nDCG@20	**5.2%**‡	**5.95%** *‡	2.61%	1.07%	2.71%	1.2%	1.25%	4.87%	1.63%
nDCG@50	**4.74%**‡	**5.02%** *‡	2.09%	0.92%	2.44%	0.94%	0.95%	4.05%	1.13%
nDCG@100	**4.09%** ‡	**4.63%** *‡	1.84%	0.79%	1.98%	0.84%	0.86%	3.82%	0.92%
AP	**68.28%**‡	**69.78%***‡	58.28%	54.28%	59.79%	53.25%	54.39%	62.02%	55.77%
AUC	**75.41%**‡	**76.35%***‡	61.51%	58.12%	52.13%	49.23%	47.42%	74.08%	56.36%
NYC									
recall@20	4.39%	**4.53%** *	3.2%	1.47%	3.51%	2.07%	2.51%	**4.78%**	2.72%
recall@50	**5.52%** ‡	**5.88%** *‡	4.84%	2.89%	4.94%	3.64%	4.21%	5.28%	4.28%
recall@100	**7.58%** ‡	**8.17%** *‡	6.26%	3.4 %	6.93%	4.29%	4.95%	6.91%	4.89%
nDCG@20	**6.72%** ‡	**6.89%** *‡	3.15%	2.73%	3.75%	2.83%	2.89%	5.92%	2.8%
nDCG@50	**5.27%** ‡	**6.06%** *‡	2.43%	2.18%	2.58%	2.34%	2.44%	5.01%	2.03%
nDCG@100	**4.76%** ‡	**4.9%** *‡	1.85%	1.34%	1.92%	1.95%	2.05%	4.52%	1.85%
AP	**69.54%**‡	**69.71%***‡	61.45%	58.73%	62.12%	59.51%	59.91%	65.29%	60.24%
AUC	**74.15%**‡	**75.92%***‡	62.38%	58.14%	63.49%	59.66%	60.15%	73.4%	61.52%

λ_f = 0 since the performance with and without social homophily of $VANF$ model show the same trends. The learning rate of SGD algorithm is kept at 10^{-6}.

6.2.2 Check-in Prediction

In this section, we compare the performance of our $VANF$ model and its extension $VANF_s$ with social homophily with several baseline models. These baseline models are also based on matrix factorization framework and they include:

- Probabilistic Matrix Factorization *PMF* [62]: PMF factorizes check-in matrix into user-latent factor and venue-latent factor matrix alone. We use the number of latent factors K = 10. We use the implementation

provided by the authors[1].

- Multi-center Gaussian Model *MGM* [12]: MGM uses multiple Gaussian distributions to model the activity centers of users. For each user, we automatically detect the clusters of check-ins by applying the non-parametric method from Blei *et. al.* [7]. We use the *MGM* implementation from Scikit-learn [65]. Each cluster is assigned as an activity center of a user. The α parameter of *MGM* which controls the weight of high frequent check-ins venues is set to default value $\alpha = 0.2$.

- Fusion Framework *PMF-MGM* [12]: PMF-MGM combines matrix factorization and *MGM*. It is reported to outperform *PMF* and *MGM* models. The probability of a user visiting a venue is determined by fusing the user's preference on that venue (returned by *PMF*) and the probability of if he/she will visit that place (returned by *MGM*).

- Matrix Factorization with Neighborhood Influence *N-MF* [35]: N-MF explores the characteristics of geographical neighbors based on the matrix factorization framework. The authors focused on studying the *spatial homophily*. We use the number of latent features $K = 10$ and two venues are neighbors if their distance is less then a predefined threshold d. In our experiment, we set d to be 100 meters and 200 meters.

- Exposure Matrix Factorization *Expo-MF* [57]: Expo-MF incorporates the location of venues and user exposure into the modeling of check-ins behavior of users. Similar to their experiment conducted in [57], we apply K-Means to cluster venues, the location vector of each venue is its probability to each cluster. We use $K = 10$ for both the number of latent factors and the number of clusters in K-Means[2].

- Social Bayesian Personalized Rankings *SBPR* [95]: SBPR assumes that

[1] https://www.cs.cmu.edu/rsalakhu/software.html
[2] https://github.com/dawenl/expo-mf

users tend to assign higher ranks to items that their friends prefer. In our experiment, we adopt the default parameters represented in the original paper. Specifically, the number of latent feature is set to 10 and the regularization parameters of users, venues and bias are 0.015, 0.025 and 0.01 respectively.

Parameter Setting: For our experiment, we adopt a *default parameter setting*. The number of latent factors is 10 by default to compare fairly with the baselines i.e. $f = 10$. The steepness of logistic function is $a = 2.0$, the width of area is $s = 0.01$. For regularization, we use $\lambda_u = \lambda_v = 0.01$. We also test the performance of the extension $VANF_s$ with social homophily. In $VANF_s$, the regularization of social homophily is $\lambda_f = 0.01$.

Result: Table 6.2 shows the performance of our $VANF$ model and the baselines under different metrics. Recall that the larger the value of each metric, the better the model. Therefore, the most important observation which we could draw from the table is that our model with default parameter setting outperforms all the baselines in general. In **SG**, **JK** and **NYC** datasets, the performance of our methods is always better than the baselines but the performance gap between $VANF$ and the baselines in **SG** dataset is larger than that in **JK** and **NYC** datasets. The reason is that the data of **JK** and **NYC** are sparser than the one of **SG** dataset. Among the baselines, *PMF-MGM* and *Expo-MF* perform better than other baselines. It happens due to the fact that these baselines cluster venues into different groups so that they could create some area attraction effects. $VANF$ model takes one step further by incorporating the neighborhood competition effect. From the results, we conclude that the impact of neighborhood competition is crucial in understanding the visitation of users in LBSNs.

From Table 6.2, we observe that using *social homophily* actually improves the performance of our model since the performance of $VANF_s$ is better than that of $VANF$ in the **SG**, **JK** and **NYC** datasets. The second observation

is that the improvement with *social homophily* is more significant in **JK** and **NYC** dataset than in **SG** dataset. For example, in **SG** dataset, *social homophily* helps us enhance 6.13% on average. The improvement in **JK** dataset is 12.03%. The reason could be that **JK** and **NYC** is sparser than **SG** so the additional information including to **JK** or **NYC** improve accuracy more significantly than the denser one (i.e. **SG** dataset).

The performance of SBPR depends heavily on the social networks of users. It is therefore not a surprise that its performance in the three datasets is not better than Expo-MF which focuses more on location of venues. Specifically, among the three datasets, **NYC** has the highest ratio of social connections and total pairs of users (0.004%) but this ratio mentioned in the original paper [95] is at least two times larger (0.01%). The reason could be that users in LBSN networks focus more on spreading their visitation than building social connection.

Significance Test: We further apply the hypothesis testing to examine if the improvement of our model is actually significant over the baselines. Since we have many baselines, we only compare the performance of $VANF$ and $VANF_s$ with the best baseline (i.e. Expo-MF). In this case, the *null hypothesis* is that the performances of our models (i.e. $VANF$ and $VANF_s$) and the baseline are not different while *alternative hypothesis* is that our models are significantly better than the baseline. To verify the hypothesises, we apply pair t-test [33] to compare the result of each metrics of $VANF$ and $VANF_s$ to the selected baseline. From the result in Table 6.2, we show that $VANF$ and $VANF_s$ are significantly better than the best baseline (i.e. Expo-MF) in most of the cases. For *recall*@20 in **NYC** dataset, the significance test fails to verify Expo-MF is better than $VANF$ and $VANF_s$ models. Particularly, the p-value of the test is 0.07 so the superior performance of Expo-MF is not significantly better than $VANF$ and $VANF_s$ models. Moreover, we also apply the above significance test to illustrate if social homophily actually improves

the performance of $VANF$ and $VANF_s$ models. Particularly, the *null hypothesis* is that the performance of both VAN and VAN_s models are not different while the *alternative hypothesis* is that VAN_s is significantly better than VAN model. As shown in Table 6.2, using social homophily helps us improve the performance of VAN model significantly.

6.2.3 Check-in Prediction for Cold Start Users

In this section, we evaluate $VANF$ and $VANF_s$ for cold start users who do not have many check-in records in our datasets.

Setup: In this experiment, we keep the same test set but in the training set, we define a user to be a cold start user if he/she has no more than 5 check-ins. The remaining users are removed from the training sets.

Parameter Settings: In this experiment, we keep the default parameter setting of $VANF$ and $VANF_s$ as described in Section 6.2.1. For the baselines, we use the parameter as described in the previous experiment.

Result: Table 6.3 shows the performance of our models and the baselines. In most of the cases, the performances of $VANF$ and $VANF_s$ are better than the performances of the baselines. We have one exception of AUC in **JK** dataset when Expo-MF outperforms $VANF$ model by a small margin. In this experiment, we also observe that Expo-MF is the best among the baseline models. For this reason, we apply the significance test between our models (i.e. $VANF$ and $VANF_s$) and Expo-MF to check if our models are significantly better than the best baseline. Moreover, we also test the significance of improvement of adding social homophily by comparing $VANF$ and $VANF_s$. From the result shown in Figure 6.3, we find that $VANF$ and $VANF_s$ are significantly better than Expo-MF. Moreover, adding social homophily actually improves the performance of model. For the exception of AUC for **JK**, we also apply the statistical test but could not find Expo-MF perform significantly better than $VANF$ and $VANF_s$.

Table 6.3: Check-in Prediction Task (Cold start Users). We boldface the best result for each performance measures. The parameters $a = 2.0$, $s = 0.01$, $f = 10$, $\lambda_u = \lambda_v = 0.01$ and $\lambda_f = 0.01$ for $VANF_s$. The symbol $*$ indicates that $VANF_s$ performs significantly better than $VANF$ while ‡ indicates the superiority of $VANF$ or $VANF_s$ over the best baseline according to significance testing.

Metrics	$VANF$	$VANF_s$	PMF	MGM	PMF-MGM	N-MF		Expo-MF	SBPR
						100m	200m		
SG									
recall@20	**7.09%** ‡	**7.92%** *‡	1.05%	0.91%	1.58%	0.5 %	0.51%	4.2%	1.55%
recall@50	**8.81 %**‡	**9.06%** *‡	1.46%	1.13 %	1.91%	0.55%	0.62%	5.9%	2.17%
recall@100	**8.94%**‡	**9.65 %***‡	2.9 %	1.87 %	2.95%	0.71%	0.75%	6.75%	4.87%
nDCG@20	**7.13%** ‡	**8.9%** *‡	2.21%	1.57%	2.35%	0.98%	1.1 %	5.87%	1.82%
nDCG@50	**6.44%** ‡	**7.32%** *‡	1.84%	1.06%	1.95%	0.52%	0.78%	4.19%	1.13%
nDCG@100	**5.08%** ‡	**6.13%** *‡	0.86%	0.45%	1.07%	0.5 %	0.56%	3.39%	0.87%
AP	**65.91%**‡	**67.41%***‡	55.18%	53.12%	58.58%	50.75%	52.37%	61.78%	57.29%
AUC	**67.18%**‡	**69.79%***‡	52.18%	51.14%	53.09%	51.45%	53.91%	63.46%	58.45%
JK									
recall@20	**3.52%** ‡	**4.18%** *‡	1.03%	0.93%	1.34%	0.67%	0.72%	2.86%	1.37%
recall@50	**4.45%** ‡	**5.73%** *‡	1.27%	1.02%	1.96%	0.93%	0.98%	3.42%	2.31
recall@100	**4.96%** ‡	**6.54%** *‡	1.88%	1.25%	2.37%	1.71%	1.83%	4.04%	3.28%
nDCG@20	**4.06%** ‡	**4.67%** *‡	1.02%	0.98%	1.24%	0.82%	0.93%	3.67%	1.31%
nDCG@50	**3.63%** ‡	**3.88%** *‡	0.95%	0.74%	1.03%	0.71%	0.81%	2.54%	1.03%
nDCG@100	**3.16%** ‡	**3.25%** *‡	0.88%	0.58%	0.91%	0.68%	0.7%	2.01%	0.92%
AP	**62.25%**‡	**64.51%***‡	53.17%	52.18%	53.58%	52.25%	52.94%	60.71%	55.48%
AUC	61.87%	**64.35%***	51.28%	52.72%	53.39%	51.92%	51.46%	**62.04%**	56.84%
NYC									
recall@20	**3.89%**‡	**4.15%***‡	1.32%	1.05%	1.48%	1.06%	1.2 %	2.51%	2.77%
recall@50	**4.55%**‡	**4.78%***‡	1.59%	1.3 %	1.68%	1.4 %	1.77%	3.17%	3.71%
recall@100	**5.61%**‡	**5.83%***‡	1.84%	1.42%	1.91%	1.74%	1.89%	4.54%	4.53%
nDCG@20	**3.66%**‡	**3.78%***‡	1.2 %	1.01%	1.57%	1.2 %	1.24%	2.62%	2.83%
nDCG@50	**2.85%**‡	**3.04%***‡	1.05%	0.96%	1.09%	1.1 %	1.12%	2.00%	2.32%
nDCG@100	**2.15%**‡	**2.58%***‡	0.92%	0.83%	0.98%	0.99%	1.09%	1.87%	2.01%
AP	**55.21%**‡	**58.91%***‡	51.15%	49.77%	51.72%	50.17%	50.23%	52.43%	53.67%
AUC	**52.12%**‡	**54.76%***‡	50.14%	48.66%	50.21%	50.25%	50.3 %	51.11%	51.21%

From Tables 6.3 and 6.2, our models and the baselines perform worse for cold-start users than for normal users so we can conclude that cold-start users are hard to predict. The reason is that the data of cold-start users is much sparser than normal ones so we do not have much data for the learning part. The prediction performance of our model for cold-start users is better than the baselines in general.

As $VANF$ and $VANF_s$ are very similar and share similar performance trend, we will study the impact of parameter settings to $VANF$ model only in the following subsections. $VANF$ is a simpler model with less parameters so it is easier for parameter tuning.

6.2.4 Tuning The Steepness Parameter

In this section, we seek to understand the role of steepness of Logistic function in modeling check-ins and its use in check-in prediction task. We try out different steepness values and observe its impact to our model performance. In this set of experiments, we only involve VANF model.

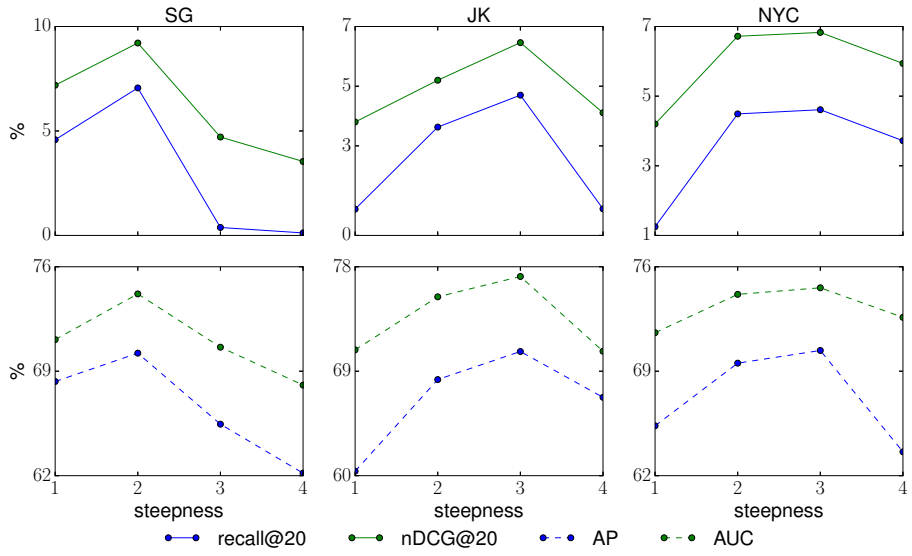


Figure 6.3: Performance of check-in prediction task of VANF model in **SG**, **JK** and **NYC** datasets with different values of steepness.

Parameter Setting: In this experiment, we vary the steepness variable a from 1.0 to 4.0 with a step size of 0.1 while keeping default values for the remaining parameters.

Result: Figure 6.3 shows the performance of VANF model with different steepness values. The best performance occurs when the value of steepness a = 2.0 for the **SG** and a = 3.0 for both **JK**, **NYC** datasets. Since a = 2.0 yields reasonably good results for all the three datasets, using this setting as default is reasonable. We also observe that the performance of VANF model degrades with larger a settings. The reason is that larger steepness values make Logistic function behaves like an indicator function which no longer nicely models the probability of competition among venues.

6.2.5 Tuning The Regularization Parameters

In this section, we try to figure out the impact of regularization parameters in modeling movement of users through check-in prediction task. To achieve the goal, we try out different values of regularization parameters. In this set of experiments, we only involve VANF model.

Parameter Setting: In this experiment, we keep the value of λ_u equal to that of λ_v since we do not want to bias to user or venue features. Recall that λ_u and λ_v are regularization parameters for the latent features of users and venues respectively. Then, we tune the values of them within the range 0 and 1 while keeping default values for the remaining parameters.

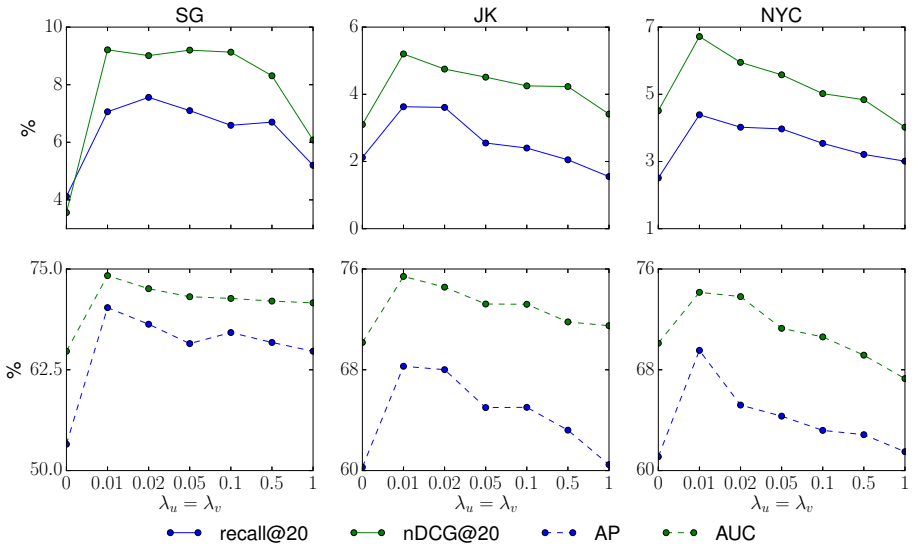


Figure 6.4: Performance of check-in prediction task of VANF model in **SG**, **JK** and **NYC** datasets with different value of regularization parameter.

Result: Figure 6.4 shows the performance of $VANF$ model for the three datasets **SG**, **JK** and **NYC** with different metrics. From the figure, we observe that without regularization (i.e. $\lambda_u = \lambda_v = 0$), the performance of $VANF$ does not perform well while increasing the value of regularization parameter also harms our model. From the figure, we can observe that selecting $\lambda_u = \lambda_v = 0.01$ yields good check-in prediction results for all the three datasets. This result suggests that our default parameter setting is reasonable.

6.2.6 Choice of Area Width

In the earlier experiments, we have adopted a fixed area width setting, i.e. s = 0.01. To understand how this setting affect the performance of VANF model, we now vary s between 0.02 to 0.002 while keeping default settings for the remaining parameters.

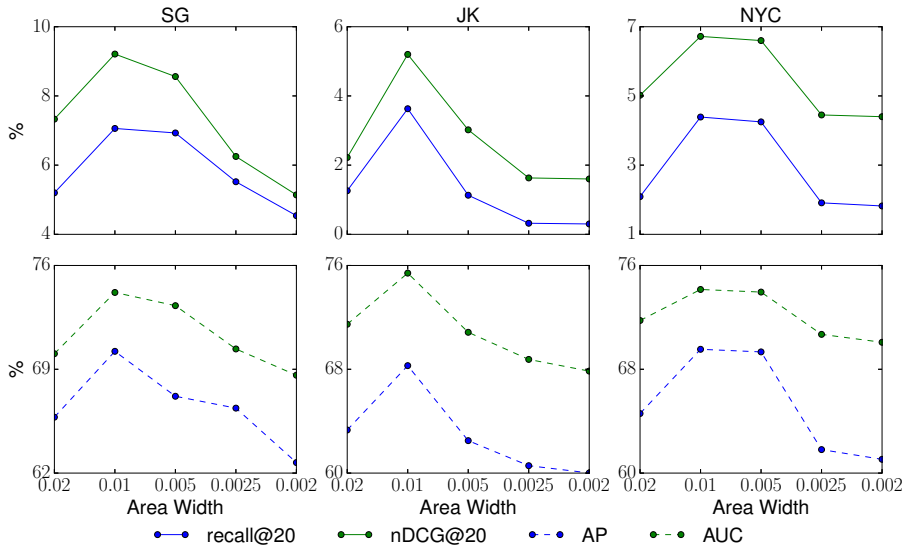


Figure 6.5: Performance of check-in prediction task of VANF model in **SG**, **JK** and **NYC** datasets with different value of area width.

Result: Figure 6.5 shows very similar performance for **SG**, **JK** and **NYC** datasets. $VANF$ model shows poorer results across different performance measures when s = 0.02 but peaks at s = 0.01 for the three datasets. Beyond s = 0.01, the performance decreases. From the result, we conclude that using s = 0.01 yields the best performance. In fact, when s is very small, each area may contain zero or one venue. Hence, the effect of area attraction is eliminated making the prediction less accurate.

6.2.7 Area Boundary Shift

In this section, we verify the robustness of our model as we shift the area boundary without changing the area size.

Parameter Setting: Recall that we create areas by dividing the city into

grid cells of equal width. The boundaries of areas are defined by vertical and horizontal lines sharing the same longitudes and latitudes respectively. Since the choice of these boundary lines can change, we would like to know if shifting the grid cells could affect the performance of VANF model. We use $VANF_x$ and $VANF_y$ to denote our model if grid cells shift 0.005 degree along latitude and longitude axes respectively. Finally, $VANF_{xy}$ is the model that shifts 0.005 degree on both latitude and longitude directions. Since the move is one half of the area width, a shift in either direction will lead to the same outcome.

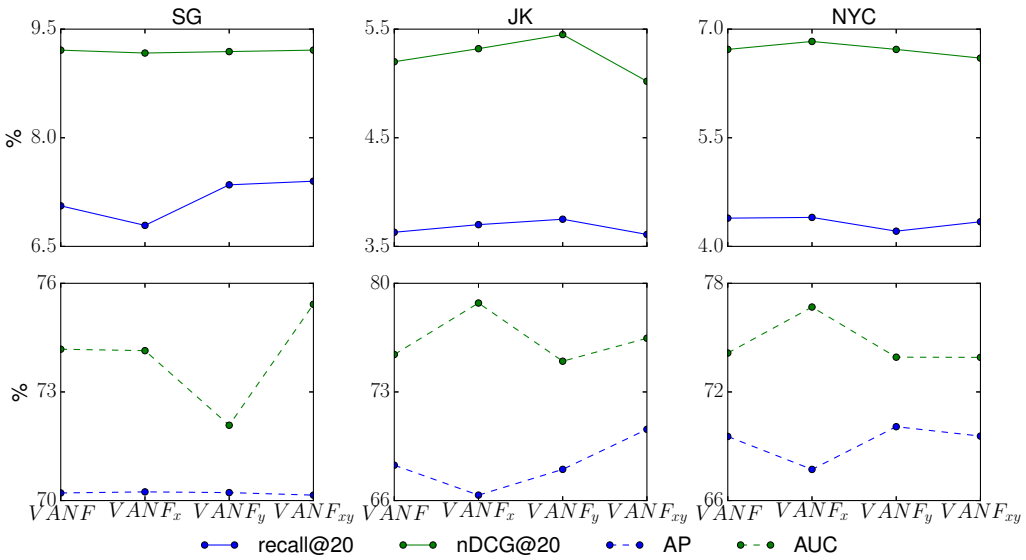


Figure 6.6: Performance of check-in prediction task of VANF model with different way of constructing areas in **SG**, **JK** and **NYC** datasets.

Result: Figure 6.6 shows the prediction result of our models using three area boundary shift settings for **SG**, **JK** and **NYC** datasets. From the result, we observe that the performance difference of $VANF_x$ and $VANF_y$ is less than 5% compared to the one of the original $VANF$ model. The performance difference between $VANF_{xy}$ and $VANF$ model is 4.6%. Therefore, we conclude that $VANF$ model is robust against different ways of area construction.

6.2.8 Venue Ranking

Other than evaluating models in check-in prediction task, we now compare the ranking of venues derived from the $VANF$ model with some known user

Table 6.4: Top 10 venues given by VANF model in **SG** dataset when $a = 2.0$, $s = 0.01$, $\lambda_u = \lambda_v = 0.01$, $\lambda_f = 0$ and the number of latent feature is 10.

Rank	Venue Name	# Check-in	# check-in users	Foursquare score	$score_v$
1	Changi International Airport	10385	5990	9.0	185.01
2	Nex	4899	1716	6.8	113.08
3	VivoCity	5456	2901	8.9	108.05
4	Jurong Point	3814	1272	7.4	98.5
5	AMK Hub	2866	1065	6.7	78.71
6	Universal Studios Singapore	3015	2415	9.3	72.23
7	ITE College East	3065	363	-	67.78
8	Compass Point	2877	706	6.1	62.72
9	Woodlands Checkpoint (Causeway)	3152	1562	-	62.54
10	Cineleisure Orchard	6470	2328	7.8	62.23

provided venue ranking. The purpose is to find out how well $VANF$ model could generate venue ranking similar to user generated venue ranking. We also compare the ranking similarity with that between other baseline models and user generated venue ranking. In this section, the user generated venue ranking comes from Foursquare score. It is a venue specific score derived by aggregating user feedback (e.g. number of likes, dislikes and tips) to the venue.

Parameter Setting: We use the default parameter setting to evaluate $VANF$ in this experiment. Due to our lack of knowledge about local language in **JK** dataset and identifiable information (i.e. the names of venues) regarding check-ins in **NYC** dataset, we only apply this task to the **SG** dataset.

Result: In the case of $VANF$ model, we compute the score of a venue v: $score_v = \sum_i p_{iv}$. Recall that p_{iv} is the probability of user i interested in venue v; hence, taking the sum over all users captures the overall interest on venue v. We then rank venues by their $score_v$'s. Table 6.4 depicts the top 10 venues that returned by VANF model. The topmost ranked venue is Changi International Airport which is a world's best airport with more than 50 million passengers per year[3]. The remaining top venues are prominent shopping malls (e.g. Nex, VivoCity, Jurong Point, AMK Hub and Compass Point), theme parks (e.g. Universal Studios Singapore), immigration checkpoint (e.g. Woodlands Checkpoint) and large education institution (e.g. ITE College East).

Ideally, we want the VANF model ranking of venues to be compared against

[3]http://www.changiairport.com/content/cag/en/aboutus.html?tab=2017

the *Foursquare score*[4]. However, not all venues in **SG** dataset have Foursquare scores. For example, Woodlands Checkpoint and ITE College East venues do not have Foursquare score (see Table 6.4). For this reason, we select only venues whose Foursquare scores are available and calculate the Pearson correlation with $VANF$'s venue ranking. The Pearson correlation score of 0.13 suggests that $VANF$ has positive correlation with Foursquare score. In other words, we can conclude that our ranking is reasonable. To quantify our ranking further, we also calculate the Pearson correlation between other models (PMF and N-MF) and Foursquare score. For PMF, the score of each venue j is $score_j^{PMF} = \sum_i U_i V_j$ and for N-MF, $score_j^{N-MF} = \sum_i \hat{R}_{ij}$ where $\hat{R}_{ij}$ is the predicted check-ins between user i and venue j by N-MF. As shown in Table 6.5, the venue ranking from $VANF$ model has the highest Pearson correlation suggesting that it performs better than other baselines by correlation with Foursquare score. Table 6.5 depicts the Jaccard similarity score between top-k ranked venues by Foursquare score and those returned by each model. The higher the value of $Jaccard@k$, the more similar the model is to Foursquare score. Specifically, suppose s_{FS}^k is the set containing top-k venues by Foursquare score and s_x^k is the set of top-k venues by model x. The Jaccard similarity score between them is $Jaccard@k = \frac{|s_{FS}^k \cap s_x^k|}{|s_{FS}^k \cup s_x^k|}$. In our experiment, we choose 20, 50 and 100 as the value of k. From Table 6.5, we observe that the Jaccard similarity score between VANF model and top venues of Foursquare score is higher than other baselines. Hence, we conclude that $VANF$ model performs better than other baselines in order to rank venues.

6.2.9 Empirical Findings and Case Studies

Finally, in this section, we present several empirical case examples to illustrate the characteristics of the VANF model using the **SG** dataset. For simplicity, we use the default parameter settings to train the $VANF$ model. In the first

[4]https://support.foursquare.com/hc/en-us/articles/201109274-Place-ratings

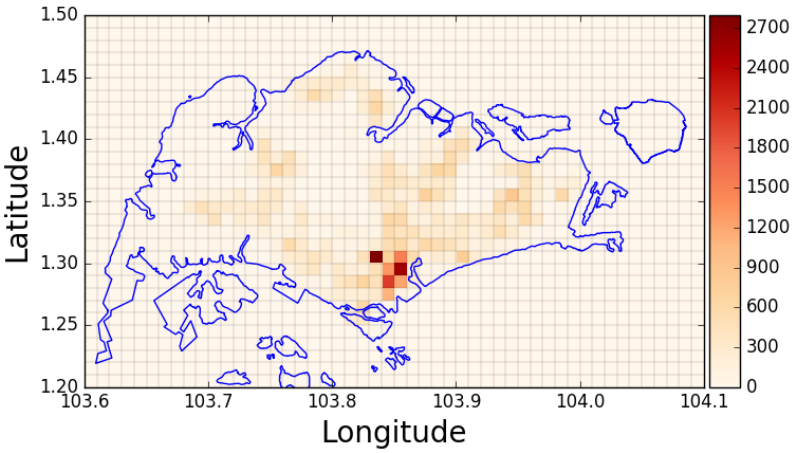
Table 6.5: Pearson Correlation and Top-k Jaccard Coefficient with Foursquare Venue Score Ranking. The best performing results are boldfaced.

Metric	VANF	PMF	N-MF	
			100m	200m
$Jaccard$@20	**8.1%**	2.6%	2.6%	2.6%
$Jaccard$@50	**11.1%**	2.1%	5.3%	7.5%
$Jaccard$@100	**14.2%**	5.3%	9.3%	8.1%
Pearson correlation	**0.13**	0.07	0.10	0.11

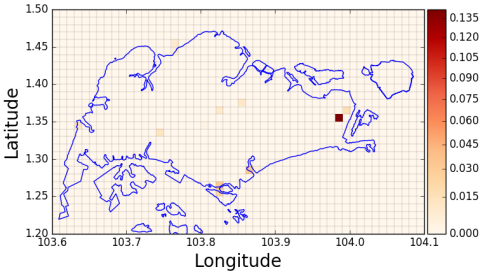
study, we examine the latent factors learned by the VANF model. Each latent factor is represented by the most representative venues. In the second study, we examine the attractiveness of areas derived by the VANF model and compare this with some simple approaches. The final study focuses on showing the competition among venues within each area to win check-ins from users.

Latent Factors: In the first study, we show the latent factors of the learned VANF model and their most representative venues in Table 6.6. The most representative venues of a latent factor are those venues v with largest $V_v[t]$ values where V_v is the latent feature vector of venue v and t is the index corresponding to the latent factor. Our findings found several latent factors related to specific location regions or groups of similar type venues. For example, the latent factors 3, 4, 7 and 8 are related to specific location regions. Particularly, latent factor 3 is represented by venues in the east of the city. Latent factors 4 and 7 cover the Orchard and City Hall shopping area respectively. Latent factor 8 is represented by subway stations. Several latent factors are related to different venue types. For example, latent factors 1, 2 and 5 are mainly shopping venues, hotels and night clubs respectively. Latent factor 10 are venues frequently visited by youths. The remaining latent factors 6 and 9 are unfortunately too noisy for interpretation. On the whole, these latent factors appear to carry reasonable meaning reflecting the different types of venues that users may be interested to visit.

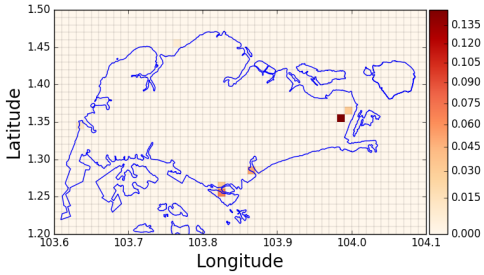
Area Attraction: In the second study, we plot the area attractiveness values derived by the VANF model in Figure 6.7a. The attractiveness of an area is



(a) **Area attractiveness.**



(b) **Area attractiveness vs check-in count.**



(c) **Area attractiveness vs user count.**

Figure 6.7: Heat map of area attractiveness returned by VANF model and its comparison with check-in count and user count using **SG** dataset.

derived by aggregating the preference of all users to this area i.e. $\sigma_{a_v} = \sum_{i \in U} \sigma^i_{a_v}$. The larger the attractiveness value, the darker the area is shaded. Figure 6.7a shows that the high attractive areas are distributed in the downtown area located in the central south of the Singapore island. We now contrast area attractiveness values with area-specific check-in counts and user counts in Figures 6.7b and 6.7c respectively. In these two figures, we normalize the attractiveness of each area by the maximum attractiveness of all areas. We also apply the similar procedure to normalize the check-in count and user count of each area. We then compute the difference between normalized attractiveness and normalized check-in count (or normalized user count) and assign shade intensity accordingly as shown in Figures 6.7b and 6.7c respectively. The two figures show that area attractiveness is very different from check-in count and user count in one specific area in the East of Singapore (indicated by dark

Table 6.6: Top 10 venues of each topic given by VANF model in **SG** dataset with $a = 2.0$, $s = 0.01$, and $f = 10$.

Topic 1 *Shopping Malls*	**Topic 2** *Hotels*	**Topic 3** *East of Singapore*	**Topic 4** *Orchard area*
Marina Bay Financial Centre	The Fullerton Hotel	Temasek Polytechnic (TP)	Takashimaya S.C.
Plaza Singapura	Swissôtel The Stamford	Changi City Point	313@Somerset
The Shoppes At Marina Bay Sands	National Library Building	Pub Glassy	ION Orchard
The Cathay	Concorde Hotel	Tampines Bus Interchange	The Paragon
Velocity	Bugis MRT Interchange	Nex	Mandarin Orchard
Chinatown Point	Grand Hyatt	St. Gabriel's Secondary School	Chambre de Louie
Great World City	Wisma Atria	Geylang West Community Club	H&M
The Central	Clarke Quay	Bugis Street	Ippudo
United Square	Strand Hotel	Blk 71 Bedok South Road	Spize River Valley
Liang Court	Citylink Mall	Liang Court	Ngee Ann City
Topic 5 *Night clubs*	**Topic 6** *Unknown*	**Topic 7** *City Hall area*	**Topic 8** *Locations around subway stations*
Club V5	313@Somerset	Nanyang Academy of Fine Arts	Marina Square
Helipad	Marina Bay Sands Casino	Marina Square	313@Somerset
Zouk	Kaplan City Campus	Bugis Junction	Cineleisure Orchard
Club Nexus	Cineleisure Orchard	City Hall MRT Station	Golden Village
Cathay Cineplex	Funan DigitaLife Mall	Golden Village	Bugis+
Strictly Pancakes	Novena MRT Station	Sin Thai Hin Building	Blk 639 Rowell Road
Liang Court	Starbucks	Raffles City Shopping Centre	Far East Plaza
Playhouse	Clarke Quay	MINK	Plaza Singapura
ZIRCA Mega Club	Zouk	Hotel Ibis	FairPrice Finest
Alfresco Gusto Italian Bistro	Marina Mandarin	Lau Pa Sat Festival Market	City Hall MRT Inter
Topic 9 *Unknown*	**Topic 10** *Youth-related Venues*		
ION Orchard	Stereo Music Store		
Raffles City Shopping Centre	Filmgarde Cineplex		
Tanjong Pagar MRT Station	Starbucks		
Funan DigitaLife Mall	Volcano Cybercafe		
Orchard Central	Bon Riche @ North Br		
Fitness First	Orchard MRT Station		
Cold Stone Creamery	Plaza Singapura		
Planet Paradise Thai Disco	Fitness First		
Paris Baguette Café	*SCAPE Flea Market		
Esplanade - Theatres On The Bay	Rebel Boutique Club		

shaded area in the figures). This area covers Changi airport which is not assigned very high attractiveness value but is known to be highly popular among the tourists and locals. This is a reasonable outcome since most users do not really like the airport and its neighboring venues (they are more likely to visit the airport for the purpose of making overseas trips.), unlike venues in the downtown areas.

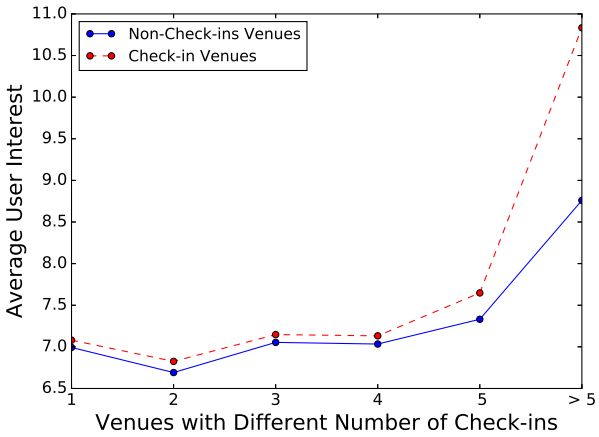


Figure 6.8: The correlation of venues with different number of check-ins and the interest of users in their most attractive areas using **SG**.

Neighborhood Competition: To show neighborhood competition within an area, this study looks into users selecting the interesting venues in the area to perform check-ins and thus creating competition among the venues. We simplify this analysis by focusing on the most favorite area of each user. The same analysis can also be applied to the less favorite areas.

For a given user i, we divide the venues in his most favorite area into different bins according to the popularity of these venues. The popularity bins cover 1, 2, 3, 4, 5 and above 5 check-ins from all users respectively. Within each bin, user i may perform check-ins on only a subset of venues from the bin. We want to show that the venues gaining the check-ins are more likely the ones winning the interest of user i. In Figure 6.8, we thus show the average user interest on these two subsets of venues for each bin of venues sharing the same popularity. The average interest of users on their visited (or unvisited) venues for each bin is computed as $\frac{1}{|U|}\sum_{i \epsilon U}\frac{1}{|\text{bin}_k^i|}\sum_{v \epsilon \text{bin}_k^i} U_i^T V_v$ where U is the set of users and bin_k^i is the set of venues with k check-ins such that user i has visited (or not visited) these venues. As shown in the figure, venues which interest users are more likely to be visited than the ones users are not interested given the same popularity.

Chapter 7

Modeling Neighborhood Competition with Spatial Homophily in Check-in Behavior

Neighborhood competition and *venue-aspect spatial homophily* effects together create the geographical neighborhood influence of venues to the check-in behavior of users. In this chapter, we capture them both under the matrix factorization-based method to understand their impact. We first give the brief overview of matrix factorization. We then introduce our proposed models that incorporate *venue aspect spatial homophily* and *neighborhood competition* effects as well as *social homophily*. We use term *spatial homophily* to refer to *venue aspect spatial homophily* henceforth. The next section sketches the method to learn parameters of the proposed model. Finally, we conduct some experiments on check-in prediction task to prove the superiority of our model [24].

7.1 Preliminaries

Our proposed model is built upon matrix factorization technique [43]. In matrix factorization, the check-in count matrix is factorized into user-specific matrix and venue-specific matrix. Formally, we assume that $R \in \mathbb{R}^{m \times n}$ is the check-in count matrix where R_{ij} is the number of check-ins user i performs on venue j. R_{ij} is undefined when user i does not perform any check-ins on venue j. m and n are the number of users and number of venues respectively. We then factorize R into two matrices $U \in \mathbb{R}^{f \times m}$ and $V \in \mathbb{R}^{f \times n}$ which satisfy $R \approx U^T V$. Therefore, the predicted number of check-ins between any pair of user i and venue j is

$$\hat{R}_{ij} = U_i^T V_j \tag{7.1}$$

where V_j represents the latent features or intrinsic characteristics of venue j such as quality, location of venue j while U_i is the vector of user i's preferences over these latent features. More notations and their meanings are shown in Table 7.1.

Nevertheless, users have some biases when performing check-ins to venues. Some users are eager to perform check-ins generating many check-ins at each visited venue while others are selective generating zero or very few check-ins. Similarly, venues also have some degree of biases because of their locations or amounts of advertisement. Hence, we represent these biases as b_i and b_j which are incorporated into the model together with a global bias μ as shown below [42]:

$$\hat{R}_{ij} = \mu + b_i + b_j + U_i^T V_j \tag{7.2}$$

Learning the latent parameters is an optimization problem as follow:

$$\min_{U_*,V_*,b_*} \sum_{(i,j)\in\mathcal{K}} (R_{ij} - \hat{R}_{ij})^2 + \lambda_1(\|U_i\|^2 + \|V_j\|^2) + \lambda_2(b_i^2 + b_j^2)$$

where λ_1 and λ_2 are regularization parameters to avoid overfitting. To learn the parameters, stochastic gradient descent (SGD) [10] is usually adopted.

Geographical Neighborhood Matrix Factorization (N-MF). Hu *et. al* [35] incorporated geographical neighborhood influence defined by the average of extrinsic characteristics of neighbors. Formally, Equation 7.2 becomes

$$\hat{R}_{ij} = \mu + b_i + b_j + U_i^T(V_j + \frac{\beta}{|N_j|}\sum_{k\in N_j} Q_k) \tag{7.3}$$

where N_j denotes the neighbors of venue j, and Q_k is the extrinsic characteristics of neighbor k. In this model, also known as Geographical Neighborhood Matrix Factorization (N-MF), the extrinsic characteristics Q_k of a venue k share the same dimension as its intrinsic characteristics V_k but the former is meant for characteristics noticeable by visitors. In this paper, we extend N-MF further to incorporate neighborhood competition.

7.2 Extended Neighborhood Matrix Factorization

In Section 3.5.2, we show that the check-ins of each venue are affected by *spatial homophily* and *neighborhood competition* effects. Hence, we propose *Extended Neighborhood Matrix Factorization (EN_MF)* model to include the two effects to check-in behavior. We extend Equation 7.3 as follow:

$$\hat{R}_{ij} = \mu + b_i + b_j + U_i^T V_j + \frac{\beta}{|N_j|}\sum_{k\in N_j} G_{ijk} U_i^T Q_k \tag{7.4}$$

In *EN_MF*, we assume that the size of N_j is identical for any venue j. In Equation 7.4, Q_k denotes the extrinsic characteristics of venue k which is

Table 7.1: Table of Notations.

Notation	**Meaning**
N_j	set of neighbors of venue j
$\mathcal{K}$	set of user-venue pairs with known check-ins
F	set of user-friend pairs
R_{ij}, $\hat{R}_{ij}$	Observed and predicted numbers of check-ins of user i to venue j, respectively
μ	Mean of all known R_{ij} check-ins
b_i, b_j	Biases of user i and venue j respectively
U_i	Latent vector of user i
V_j/Q_j	Intrinsic/Extrinsic characteristic vector of venue j
β	Parameter to control the effect of neighborhood venues
α	Relative weight between spatial homophily and neighborhood competition

a neighbor of venue j and its product with U_i contributes to the number of check-ins between user i and venue j. First of all, we need to explain that Q_j has the same number of latent dimensions as V_j. Each Q_{jt} element captures the ability of a venue j to bring check-ins from users interested in t-th latent factor to its neighbors. G_{ijk} denotes the neighborhood influence weight which is defined to be a combination of venue j winning over the neighboring venue k (*neighborhood competition*), and similarity with the neighboring venue k (*spatial homophily*) as user i chooses venue j over its neighbor k for check-ins. Formally, G_{ijk} is:

$$G_{ijk} = \alpha\sigma(U_i^T Q_j > U_i^T Q_k) + (1-\alpha)sim(j,k) \tag{7.5}$$

The two parameters β ($\beta > 0$) and α ($\alpha \in [0,1]$) in Equations 7.4 and 7.5 are:

- β controls the geographical neighborhood influence of neighboring venues.
- α is the tradeoff between spatial homophily and neighborhood competition.

$sim(j,k)$ in Equation 7.5 measures the effect of *spatial homophily* of the neighbor k of venue j to the selection of venue j by users. By including

$sim(j,k)$, our model covers the spatial homophily effect among venues. We explore $sim(j,k)$ function further by considering these following options to capture our observations in Section 3.5.2:

- *Check-in cosine similarity:* Cosine similarity between check-in counts of users of two venues j and k.
- *Distance cosine similarity:* Cosine similarity of distance of common users between venue j and venue k.

$\sigma(U_i^T Q_j > U_i^T Q_k)$ in Equation 7.5 captures the competition between venue j and its neighbor k. The intuition behind is that from the perspective of user i, the extrinsic characteristics of venue j are ranked higher than those of its neighbor k. User i therefore selects venue j to visit instead of its neighbor k. In other words, user i prefers venue j over k by comparing the extrinsic characteristics of j and k. Function σ returns the probability that user i is more attracted to venue j than k. In this work, we consider two options for function σ:

- Sigmoid function: We adopt this option from the study of personal ranking using matrix factorization [71]. Formally, $\sigma(U_i^T Q_j > U_i^T Q_k) = \frac{1}{1+\exp(-(U_i^T Q_j - U_i^T Q_k))}$
- Cumulative density function of standard normal distribution (CDF): Similar to Sigmoid function, we use CDF to map the value of $U_i^T Q_j - U_i^T Q_k$ into the range $[0,1]$.

Finally, our task is to learn the parameters U_*, V_*, Q_* and b_* through solving the following optimization problem by using gradient descent method [10]:

$$\min_{U_*,V_*,Q_*,b*} \sum_{(i,j)\epsilon\mathcal{K}} (R_{ij} - \hat{R}_{ij})^2 + \lambda_1(\|U_i\|^2 + \|V_j\|^2) + \lambda_2(b_i^2 + b_j^2) + \lambda_3\|Q_j\|^2 \quad (7.6)$$

Note: The special thing is that our model is the generalization of *N-MF* model proposed by Hu *et. al.* [35]. Specifically, if we set $\alpha = 0$ and the $sim(j,k) = 1$ for all venues j,k, then our model reduces to *N-MF* model (see Equation 7.3).

Extension incorporating social homophily(FEN_MF): Similar to [12], we model social homophily by adding a social regularizer $\lambda_f \sum_{(i,i')\in F} \|U_i - U_{i'}\|^2$ to Equation 7.6. It says that if two users i and i' have social connection, their latent features U_i and $U_{i'}$ tend to have similar values and λ_f is the parameter to control the impact of social homophily.

7.2.1 Parameter Learning

To learn the parameters of EN_MF, we apply SGD framework for matrix factorization [42]. The core component of the framework is the gradient of parameters that we want to learn. To ease reading, we use $\mathcal{L}$ to denote the half of function that we want to optimize in Equation 7.6 and $e_{ij} = \hat{R}_{ij} - R_{ij}$ so the derivatives of $\mathcal{L}$ with respect to the parameters are

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial b_i} &= \sum_{(i,j)\in\mathcal{K}} e_{ij} + b_i\lambda_2; \ \frac{\partial \mathcal{L}}{\partial b_j} = \sum_{(i,j)\in\mathcal{K}} e_{ij} + b_j\lambda_2 \\ \frac{\partial \mathcal{L}}{\partial V_{jt}} &= \sum_{(i,j)\in\mathcal{K}} e_{ij}U_{it} + \lambda_1 V_{jt} \end{aligned} \tag{7.7}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial U_{it}} = \sum_{(i,j)\in\mathcal{K}} e_{ij}\Bigg[\frac{\beta}{|N_j|}\sum_{k\in N_j}\Bigg(&\alpha U_i^T Q_k \frac{\partial}{\partial U_{it}}\sigma(U_i^T Q_j > U_i^T Q_k) \\ &+ G_{ijk}Q_{kt}\Bigg) + V_{jt}\Bigg] + \lambda_1 U_{it} \end{aligned} \tag{7.8}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial Q_{kt}} = \sum_{(i,j)\in\mathcal{K}} e_{ij}\frac{\beta}{|N_j|}\sum_{k\in N_j}\Bigg(&\alpha U_i^T Q_k \frac{\partial}{\partial Q_{kt}}\sigma(U_i^T Q_j > U_i^T Q_k) \\ &+ G_{ijk}U_{it}\Bigg) + \lambda_3 Q_{kt} \end{aligned} \tag{7.9}$$

If neighborhood competition is modeled by Sigmoid function, we have

$$\begin{aligned}
&\frac{\partial}{\partial U_{it}}\sigma(U_i^T Q_j > U_i^T Q_k) \\
&= \left(-\sigma(U_i^T Q_j > U_i^T Q_k) + \sigma(U_i^T Q_j > U_i^T Q_k)^2\right)[Q_{jt} - Q_{kt}] \\
&\frac{\partial}{\partial Q_{kt}}\sigma(U_i^T Q_j > U_i^T Q_k) \\
&= \left(\sigma(U_i^T Q_j > U_i^T Q_k) - \sigma(U_i^T Q_j > U_i^T Q_k)^2\right) U_{it}
\end{aligned} \tag{7.10}$$

In the case of modeling neighborhood competition by CDF, we have the corresponding derivatives as follow

$$\begin{aligned}
\frac{\partial}{\partial U_{it}}\sigma(U_i^T Q_j > U_i^T Q_k) &= \mathcal{N}(U_i^T Q_j - U_i^T Q_k; 0, 1)\Big(Q_{jt} - Q_{kt}\Big) \\
\frac{\partial}{\partial Q_{kt}}\sigma(U_i^T Q_j > U_i^T Q_k) &= -\mathcal{N}(U_i^T Q_j - U_i^T Q_k; 0, 1) U_{it}
\end{aligned} \tag{7.11}$$

where $\mathcal{N}(\bullet; 0, 1)$ represents the probability density function of standard normal distribution.

Parameter Learning for extension incorporating social homophily(FEN_MF): In this extension, we add the gradient of social regularizer $\lambda_f \sum_{(i,i')\in F} (U_{it} - U_{i't})$ when we compute the gradient of user i (e.g. $\frac{\partial \mathcal{L}}{\partial U_{it}}$). Specifically, we have

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial U_{it}} = \sum_{(i,j)\in\mathcal{K}} e_{ij}\Bigg[\frac{\beta}{|N_j|}\sum_{k\in N_j}\Bigg(&\alpha U_i^T Q_k \frac{\partial}{\partial U_{it}}\sigma(U_i^T Q_j > U_i^T Q_k) \\
&+ G_{ijk} Q_{kt}\Bigg) + V_{jt}\Bigg] + \lambda_1 U_{it} + \lambda_f \sum_{(i,i')\in F} (U_{it} - U_{i't})
\end{aligned} \tag{7.12}$$

7.3 Experiments

In this section, we describe our experiments on Foursquare datasets to evaluate our proposed model against other baselines. Moreover, some intensive experiments are also conducted to ensure the robustness of our models in LBSN.

7.3.1 Experimental Setting

In this experiment, we evaluate the performance of our model using check-in prediction task.

Setup: We sort the check-ins of the **H_SG** and **H_JK** datasets in chronological order and divide each dataset into ten folds. For each run of experiment, we hide one fold as test set and use the remaining nine folds as training set.

We order check-ins of the **SG**, **JK** and **NYC** chronologically, and then divide the data into two parts: the first 80% is for training and the remaining 20% is for testing. There are no home location for all users in these datasets so to apply the *distance cosine similarity*, we approximate the home locations of users by deriving the centers of the mass from all check-in venues of the users.

Evaluation metric: We adopt two popular error metrics, *Mean Absolute Error* (MAE) and *Root Mean Square Error* (RMSE). The smaller the value of MAE and RMSE, the more accurate the model is. In general, RMSE penalizes more on the large errors and less on smaller ones than MAE. Suppose T is the test set containing user-venue check-in pairs (i, j)'s, the two metrics are:

$$\begin{aligned} MAE &= \frac{1}{|T|} \sum_{(i,j)\in T} |R_{ij} - \hat{R}_{ij}| \\ RMSE &= \sqrt{\frac{1}{|T|} \sum_{(i,j)\in T} (R_{ij} - \hat{R}_{ij})^2} \end{aligned} \tag{7.13}$$

We report the average MAE and $RMSE$ of all ten folds. For the ease of reading, we use MAE and $RMSE$ to refer to average MAE and average $RMSE$ respectively henceforth.

Proposed Models: Our proposed models to be evaluated are:

- $EN\_MF^{DS}_{Sigmoid}$: In this model, *distance cosine similarity* is used for *spatial homophily* and the *Sigmoid function* is adopted for *neighborhood competition.*
- $EN\_MF^{CS}_{Sigmoid}$: This model uses *check-in cosine similarity* for *spatial homophily* and *Sigmoid function* for *neighborhood competition.*

- $EN\_MF^{DS}_{CDF}$: In this model, *distance cosine similarity* is adopted for *spatial homophily* and *CDF* is to model *neighborhood competition* effect.

- $EN\_MF^{CS}_{CDF}$: This model uses *check-in cosine similarity* and *CDF* to model *spatial homophily* and *neighborhood competition* respectively.

$FEN\_MF^{DS}_{Sigmoid}$, $FEN\_MF^{CS}_{Sigmoid}$, $FEN\_MF^{DS}_{CDF}$ and $FEN\_MF^{CS}_{CDF}$ are the extension of $EN\_MF^{DS}_{Sigmoid}$, $EN\_MF^{CS}_{Sigmoid}$, $EN\_MF^{DS}_{CDF}$ and $EN\_MF^{CS}_{CDF}$ respectively by adding social homophily.

Baselines: The baseline models are described below:

- *User Mean:* To predict the number of check-ins between a user and a venue, it outputs the average number of check-ins of this user performs to a venue.

- *Bias Matrix Factorization (B-MF)*: This matrix factorization model was proposed by Koren [42]. In this model, the biases of users and venues are considered and it is briefly mentioned in Section 7.1.

- *Neighborhood influence Matrix Factorization (N-MF)*: Hu *et. al.* [35] proposed a model to incorporate only the effect of *spatial homophily*. It is the special case of our model (see Section 7.2).

Parameter Setting: We adopt a parameter setting similar to that of [35] for *EN_MF*, *FEN_MF* models and *N-MF* since it provides overall good performance for the baselines. That is, the number of latent factors is f = 20, and neighborhood importance is β = 0.8. The regularization parameters: λ_1 = 0.8, λ_2 = 0.4, λ_3 = 0.6 and λ_f = 0.01. The learning rate of SGD γ is assigned to 0.00001. Besides the above parameters, we also set α = 0.5 to give equal weights to both: *spatial homophily* and *neighborhood competition* effects. For *EN_MF*, *FEN_MF* and *N-MF*, we consider the top 10 nearest venues as neighbors of a venue since it generates a good result across multiple variants (more details in later sections).

Table 7.2: Performance of check-in prediction task. The best results are highlighted.

	H_SG		**H_JK**		**SG**		**JK**		**NYC**	
Method	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
User Mean	1.9621	17.2189	1.7530	12.7721	1.642	12.1344	1.0923	13.2112	1.232	11.389
B-MF	1.8122	15.2199	1.6892	11.2758	1.4812	11.4354	0.9873	12.8085	1.092	10.891
N-MF	1.7522	14.7212	1.4016	9.4293	1.4033	11.4266	0.9784	12.7491	0.992	10.003
$EN\_MF^{DS}_{Sigmoid}$	1.6974	14.3460	1.2475	9.2948	1.39	11.4156	0.9638	12.7005	0.979	9.821
$EN\_MF^{CS}_{Sigmoid}$	1.6975	14.3424	1.2471	9.2942	1.3872	11.4150	0.9624	12.7057	0.971	9.754
$EN\_MF^{DS}_{CDF}$	1.6965	14.3463	1.2475	9.2936	1.3899	11.4148	0.9635	12.7058	0.98	9.892
$EN\_MF^{CS}_{CDF}$	1.6964	14.3421	1.2469	9.2946	1.3873	11.4177	0.9628	12.7095	0.972	9.786
$FEN\_MF^{DS}_{Sigmoid}$	1.6957	14.3451	1.21795	**8.2367**	1.3890	11.4135	0.9633	12.6996	0.9612	9.79
$FEN\_MF^{CS}_{Sigmoid}$	1.6942	14.342	1.2172	8.3744	1.3872	11.4147	**0.9624**	**12.6953**	0.9641	9.809
$FEN\_MF^{DS}_{CDF}$	1.6959	14.346	1.2175	8.2832	1.3890	**11.4133**	0.9632	12.6970	0.9617	9.701
$FEN\_MF^{CS}_{CDF}$	**1.6941**	**14.3417**	**1.2164**	8.2789	**1.3871**	11.4150	0.9625	12.6992	**0.9604**	**9.68**

7.3.2 Experiment Results

We conduct the experiment to compare the performance of our proposed *EN_MF* and FEN_MF with several baselines. We then evaluate the impact of neighborhood size to the prediction accuracy of *EN_MF*. Next, we also tune parameter α to measure the contribution of *neighborhood competition* and *spatial homophily* to the prediction accuracy of *EN_MF*. We do not report the performance of FEN_MF on the last two experiments since its behavior is similar to EN_MF.

7.3.2.1 Check-in Prediction Task.

The performance of all the four variants of EN_MF and FEN_MF as well as the baselines on the four datasets **SG**, **H_SG**, **JK**, **H_JK** and **NYC** are listed in Table 7.2.

Firstly, all four variants of EN_MF and FEN_MF perform better than the baselines. Specifically, FEN_MF could improve up to 13.49% in MAE and 16.8% in RMSE compared to the baselines. It suggests that incorporating *spatial homophily* and *neighborhood competition* as well as *social homophily* effectively reduce prediction errors. The performance is superior than baseline models that do not consider any effects (i.e. *User Mean*, *B-MF*) or the one (i.e. *N-MF*) that incorporates only the *spatial homophily* effect. We further apply hypothesis testing to examine if our improvements are significantly bet-

ter than the baselines. Specifically, the *null hypothesis* is the performance of our methods and the baselines are not different while the *alternative hypothesis* is our methods are significantly better than the baselines. To achieve the goal, we apply the paired t-tests [33] to compare each variant of EN_MF and FEN_MF to *N-MF*. The population size in our tests is 10 (the number of folds in our experiment). Since the *p-values* of all tests are less than 0.05, we conclude that EN_MF and FEN_MF are significantly better than the baselines. Next, we also perform significant test to compare between each variant of EN_MF and the corresponding variant of FEN_MF. From the result of the test, we found that FEN_MF model variants significantly improve those of EN_MF.

Secondly, Table 7.2 shows that $FEN\_MF^{CS}_{CDF}$ has the best overall performance on the **H_SG** and **H_JK** datasets. Recall that it uses *check-in cosine similarity* and *CDF* to model the effects of *spatial homophily* and *neighborhood competition* respectively. This model produces the lowest prediction errors in both datasets except the case of RMSE in **H_JK**. Hence, using *CDF* is more appropriate for modeling *neighborhood competition* than *Sigmoid function*. Similarly, characterizing *spatial homophily* by *check-in cosine similarity* is more accurate than using *distance cosine similarity*. For the large datasets **SG** and **JK**, it is hard to find the best model.

Thirdly, the MAE and RMSE errors in Table 7.2 are higher than those reported by Hu *et. al.* [35] since they used Yelp dataset to evaluate prediction performance of *N-MF*. Specifically, *N-MF* predicts the ratings of users to venues and the ratings can obtain a discrete value from 1 to 5. In contrast, we apply *N-MF* and our models to predict the number of check-ins between users and venues and such number can be much larger than 5. Hence, the figures reported in Table 7.2 are significantly higher than the ones showed in [35].

Next, Table 7.2 shows that EN_MF performs better than *N-MF* by incorporating additional neighborhood effects. *User Mean* method does not cover

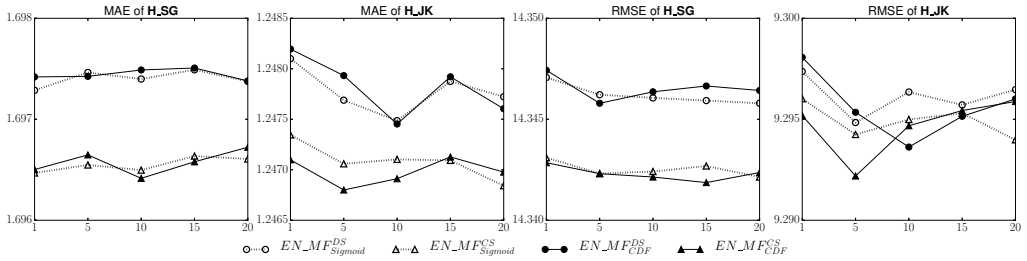


Figure 7.1: Performance of variants of EN_MF with different numbers of neighbors in **H_SG** and **H_JK**.

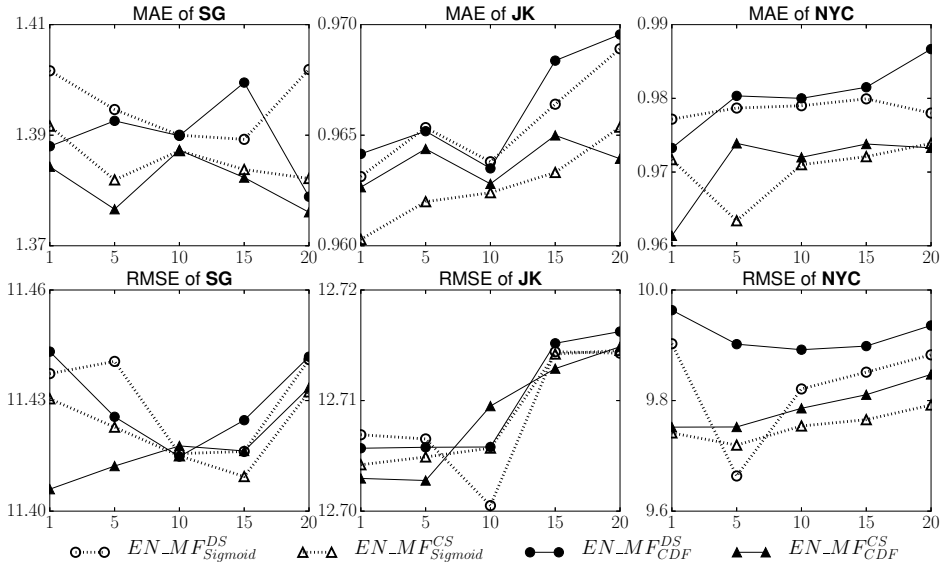


Figure 7.2: Performance of variants of EN_MF with different numbers of neighbors in **SG**, **JK** and **NYC**.

any information of venues so its results are not better than that of *B-MF* which includes the interaction between users and venues. However, *N-MF* outperforms *B-MF* because it considers spatial homophily.

Lastly, social homophily can improve the prediction performance and this phenomenon happens across all variants. However, the improvement of using social homophily is small, consistent with the result reported in previous works [12].

7.3.2.2 Choice of Neighborhood Size.

In our models, the neighbors of a venue are the top-n nearest neighbors of this venue. To measure the impact of n, we vary n to quantify the impor-

tance of neighborhood size to the prediction errors of all variants of EN_MF. Figure 7.1 depicts the finding in both **H_SG** and **H_JK** datasets. For other parameters of *EN_MF*, we use their default values. There are three useful observations from Figure 7.1.

First of all, in **H_SG**, the prediction errors of all variants of EN_MF are more stable than the ones of **H_JK** dataset and it is hard to observe the trending of our error metrics when n is varied for the dataset **H_SG**. Secondly, we can group the variants into two groups: *check-in* and *distance cosine similarity* groups since the first one usually has lower prediction errors than the other. This result is consistent on both the two datasets and both error metrics except in the case of RMSE in **H_JK**. It suggests that we should use *check-in cosine similarity* to model the *spatial homophily* of two venues to achieve smaller prediction errors. Thirdly, from **H_JK** dataset, we observe that three out of four variants of EN_MF achieve the lowest RMSE value at when number of neighbors of a venue is 5 while only $EN\_MF^{CS}_{CDF}$ obtains the lowest MAE at $n = 5$.

The reason behind the differences between **H_SG** and **H_JK** is the sparsity of **H_JK**. From Table 3.1, the number of venues of **H_JK** is one third of that of **H_SG**. Therefore, increasing the number of neighbors of a venue j is equal to the fact of considering more further away venues as neighbors of j. Consequently, it reduces the accuracy of EN_MF.

Hence, we could conclude that in datasets whose venues are dense (e.g. **H_SG**), the number of neighbors in our model does not affect the prediction performance as much as datasets whose venues are sparse (e.g. **H_JK**).

Figure 7.2 illustrates the performance of variants of EN_MF with different of neighbors on **SG**, **JK** and **NYC** datasets. The figure also shows the same trend as the performance of **H_SG** and **H_JK** so the observations above are still applied to the full datasets.

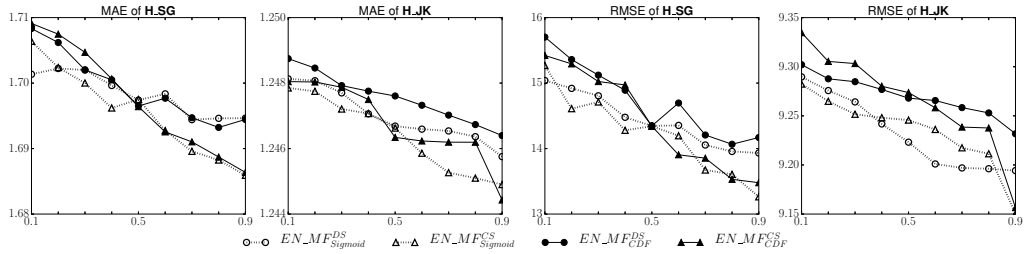


Figure 7.3: Prediction errors of variants of EN_MF with different values of α in **H_SG** and **H_JK**.

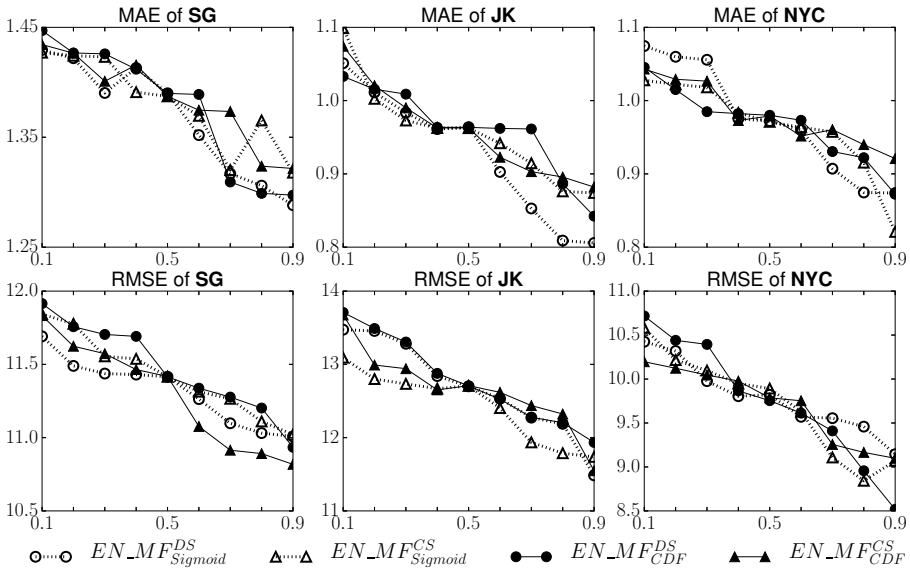


Figure 7.4: Prediction errors of variants of EN_MF with different values of α in **SG**, **JK** and **NYC** datasets.

7.3.2.3 Spatial Homophily vs Neighborhood Competition.

The role of α in Equation 7.4 is to control the impact of two effects: *spatial homophily* and *neighborhood competition*. Specifically, if $\alpha \to 0$, the effect of *neighborhood competition* is eliminated in EN_MF model. Otherwise (i.e. $\alpha \to 1$), the effect of *spatial homophily* is left out in EN_MF.

In this section, we want to quantify the influence of both effects. For that reason, we vary the value of α from 0.1 to 0.9 with step 0.1 and measure the prediction errors of EN_MF and its variants. We use the default values for other parameters during the experiment. As shown in Figure 7.3, the prediction errors of all versions of EN_MF in **H_SG** and **H_JK** reduce when we increase α. The exceptions are the cases of $EN\_MF^{DS}_{Sigmoid}$ and $EN\_MF^{DS}_{CDF}$

on **H_SG** dataset. For example, the MAE and RMSE of these two models increase when α changes from 0.5 to 0.6 but these errors drop when α increases to 0.7. However, the errors of $EN\_MF^{DS}_{Sigmoid}$ and $EN\_MF^{DS}_{CDF}$ decrease when we increase the value of α. Hence, in general, we could conclude that *spatial homophily* effect contributes less to the accuracy of check-in prediction than *neighborhood competition*. Despite this findings, the contribution of *spatial homophily* is not negligible because the worst performing in both datasets still perform better than the baselines. The other observation from Figure 7.3 is that we cannot conclude which model has the best performance since there are no clear winner among them. We repeat the same experiment in **SG**, **JK** and **NYC** datasets to check the robustness of all versions of EN_MF model. As shown in Figure 7.4, the finding is still consistent in the large datasets.

Chapter 8

PACELA: A Neural Framework for Check-in Behavior using Both Observed and Latent Attributes of Users and Venues

The recent breakthroughs in deep learning have brought about a plethora of new unsupervised and supervised learning techniques [30]. These techniques, despite their higher computation costs, are shown to yield high accuracy in prediction tasks. Given the check-in prediction challenges, it is therefore interesting to explore a deep learning or neural framework to generate better prediction results at the same time incorporating both embedding and the latent attribute features behind the various factors relevant to check-in behavior. Hence, in this chapter, we propose a neural framework that could integrate the latent attributes of users and venues to model the check-in behavior of users in LBSNs.

8.1 Proposed Model

8.1.1 Model Description

In this section, we propose a framework called Preference And Context Embeddings with Latent Attributes (PACELA).

The input of PACELA framework consists of: (a) users and their social connections; (b) venues with locations; and (c) check-ins performed by users on venues. We use N and M to denote the total number of users and venues respectively. In this paper, we define the context of a user i to be the set of users who have social connections with user i which is denoted by $\mathbf{u}_i$. We also define the context of a venue j to the set of venues that are nearby, as denoted by $\mathbf{v}_j$. The set of all check-ins is denoted by C and each check-in is a tuple $\{(u_i, v_j)\}$ representing user i has performed a check-in on venue j. From C, we can define a check-in variable y_{ij} such that $y_{ij} = 1$ if $(u_i, v_j) \in C$, and $y_{ij} = 0$ otherwise.

As shown in Figure 8.1, this framework consists of four components, namely, the two *network embedding* components for learning user context and venue context, a *latent attribute modeling* component for learning user and venue attributes, and a *neural network* component for predicting check-ins between users and venues. By instantiating these components with an appropriate model, we can realize different models for check-in behavior.

The network embedding component for user context essentially takes the user social network data and learns an embedding space. Users will be mapped into this embedding space such that users with similar context will be close to one another in this space. Similarly, the network embedding component for venue context learns an embedding space using the venue proximity network. This way, venues with similar spatial neighbors will be close to one another in the embedding space.

The latent attribute modeling component takes all check-in history data of

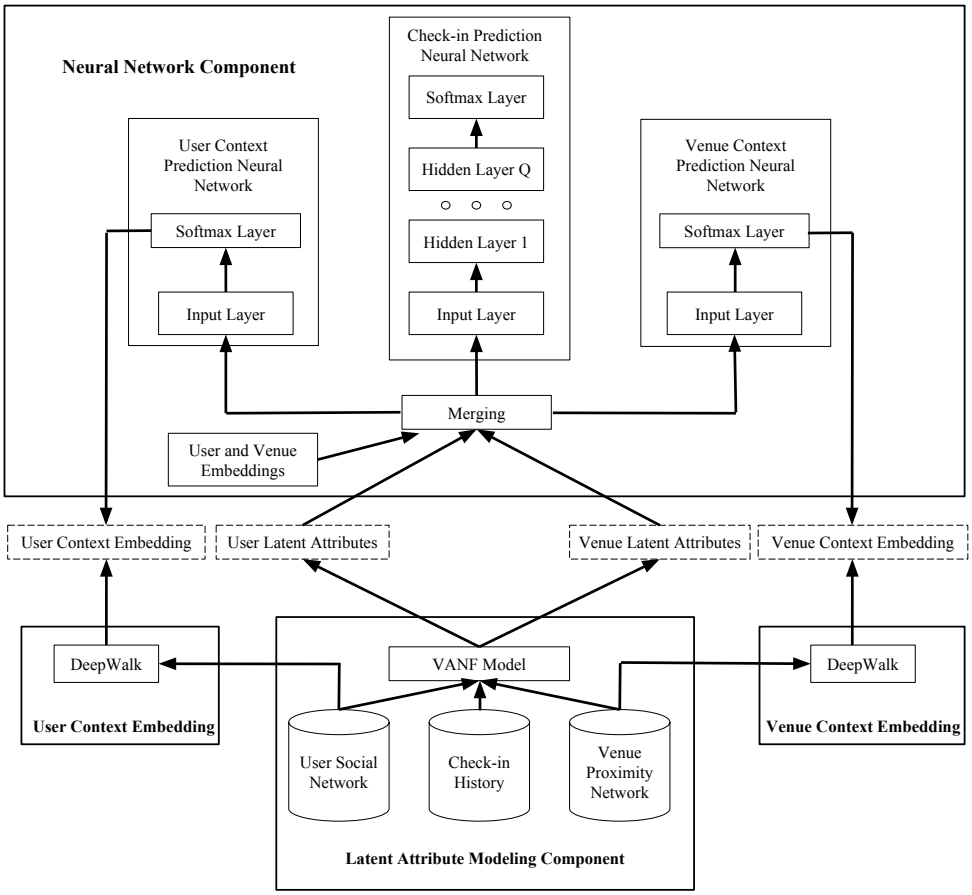


Figure 8.1: Neural Architecture of PACELA model.

users as well as the users' social networks and venues' proximity networks to learn the latent attributes of users and venues respectively.

Finally, we have the neural network component that merges all user and venue latent attributes together to predict check-ins, user context and venue context at the same time learning for each user and venue, a user embedding vector and venue embedding vector respectively. The prediction of check-ins utilizes a multi-layer perceptron network, while the predictions of user and venue context embedding utilize a single layer neural network. Particularly, we use concatenation to merge user and venue latent attributes. The reason of using concatenation over element-wise operator is that (i) vector concatenation is able to model non-linear interactions between users and venues, and (ii) vector concatenation does not require both vectors to be in the same space.

In the following, we will introduce a specific model instantiation using the

framework.

8.1.2 Model Formalization

Network Embedding Components. We use DeepWalk, a well known network embedding model, to learn the embeddings of user context and venue context [66]. DeepWalk uses random walk to establish the local information of each node in the network and learns the distributed representation vector of the node. In this paper, users form a social network and venues form a venue proximity network. We set the dimension of embedding vector of a user or venue to 64 by default. The embedding vectors of all users can then be represented by a $N \times 64$ matrix X_u. To retrieve the embedding vector of user i, we can compute $X_u^T u_i$ where u_i is represented as a one-hot vector. Similarly, for venue, we can define another embedding matrix X_v whose size is $M \times 64$ and retrieve the embedding vector of venue j by $X_v^T v_j$. For the ease of reading, we denote the representative vectors of context of user i and venue j as $\mathbf{u}_i$ and $\mathbf{v}_j$ respectively.

Latent Attribute Modeling Component. The goal of this component is to extract the latent attributes of users and venues. In this dissertation, we have chosen to extract or derive the user and venue latent attributes which are relevant to *area attraction*, *neighborhood competition* and *social homophily*. Then, the learned latent attributes are combined together for learning under the neural network component. In particular, we have chosen to use the matrix factorization model VANF proposed in Chapter 6 for deriving these latent attributes. The inputs of VANF include the social network of users, check-in history of users and venue proximity network. From these inputs, we use matrix factorization-based method to derive user and venue attributes. We denote the latent attributes of user i and venue j as u_i' and v_j' respectively.

Neural Network Component. We use a single layer neural network in PACELA to return predictions of user context of user i, and another similar

neural network for venue context of venue j. A multi-layer neural network is used to predict check-in of i on j. The predicted variables are denoted as $\hat{u_i}$, $\hat{v_j}$, and $\hat{y_{ij}}$ respectively.

These predictions are generated by the softmax layer of the three neural networks. We first describe the prediction of check-in variable y_{ij} using a multi-layer neural network $\mathcal{H}$.

$$\hat{y}_{ij} = h(E_u^T u_i, E_v^T v_j | \Theta_e, \Theta_h, u_i', v_j') \tag{8.1}$$

where Θ_e denotes the parameters of the embedding layer while Θ_h represents the parameter of preference prediction layer. Moreover, u_i' and v_j' are the vectors of latent attributes of user i and venue j respectively.

As shown in Figure 8.1, $\mathcal{H}$ has Q layers. The input layer consists of the embedding vectors of user i and venue j and their latent attributes u_i' and v_j'. Hence, we denote the input layer by $x_{ij} = [E_u^T u_i; u_i'; E_v^T v_j; v_j']$.

x_{ij} is then fed into the first hidden layer of $\mathcal{H}$ which has full connectivity between input layer and the first hidden layer, as well as full connectivity between two hidden layers. The q-th hidden layer of $\mathcal{H}$ denoted as h^q is defined as a non-linear function of its previous hidden layer h^{q-1}. Formally, we have:

$$h^q(x) = ReLU(W^q h^{q-1}(x) + b^q) \tag{8.2}$$

where W^q and b^q are the parameters of the q-th layer of $\mathcal{H}$. $h^0(x_{ij}) = x_{ij} = [E_u^T u_i; u_i'; E_v^T v_j; v_j']$. We choose the rectified linear unit $ReLU(x) = max(0, x)$ as the non-linear function.

After Q layer of computation, the prediction of check-in variable, $\hat{y}_{ij}$, can be expressed as:

$$\begin{aligned} \hat{y}_{ij} &= h_{pred}(h^Q(\cdots h^1(h^0([E_u^T u_i; u_i'; E_v^T u_j; v_j'])) \cdots)) \\ &= h_{pred}(H^Q([E_u^T u_i; u_i'; E_v^T v_j; v_j'])) \end{aligned} \tag{8.3}$$

where h_{pred} is a softmax involving logistic regression with Sigmoid function. It turns the output of $H^Q([E_u^T u_i; u_i'; E_v^T v_j; v_j'])$ from a vector form to a prediction value between 0 and 1. In other words, we have the formula:

$$\hat{y}_{ij} = S(H^Q([E_u^T u_i; u_i'; E_v^T v_j; v_j'])^T w_y) \tag{8.4}$$

where the Sigmoid function is defined as $S(x) = 1/(1 + e^{-x})$ and w_y is the parameter vector of the softmax layer.

The multi-layer neural network has two configuration parameters, Q (number of hidden layers) and R (capacity). The capacity R is the size of the last hidden layer Q, i.e., h^Q. The size of each hidden layer (except the last one) is assigned to be twice the size of the next hidden layer. Hence, for a multi-layer neural network with $Q = 4$ and $R = 2$, the size of layer q of the network is $h^q = R^{Q-q+1}$. Recall the h^0 refers to the input layer and its size is determined by the embedding vectors and latent attributes.

A single layer perceptron network is used to predict the context of user i. Again, we concatenate the embedding vector of user i with his latent attribute vector u_i'. Formally, the context prediction vector of user i is generated by

$$\hat{u}_i = S([E_u^T u_i; u_i']|\phi_{u_i}) = S([E_u^T u_i; u_i']^T \phi_{u_i}) \tag{8.5}$$

where $S(\cdot)$ is the sigmoid function that applies to each element of the given vector and ϕ_{u_i} is the parameter of the densely connected neural network for user i. Recall that $E_u^T u_i$ is the embedding vector of user u_i.

Similarly, we also a single layer perceptron to predict the context of venue j as follows.

$$\hat{v}_j = S([E_v^T v_j; v_j']|\psi_{v_j}) = S([E_v^T v_j; v_j']^T \psi_{v_j}) \tag{8.6}$$

where ψ_{v_j} is the network parameter of venue j.

Loss functions (Neural Network). The above three neural networks are jointly trained by optimizing the sum of three loss functions as follows

$$\mathcal{J} = \mathcal{J}_Y + \lambda_1 \mathcal{J}_{C_U} + \lambda_2 \mathcal{J}_{C_V} \tag{8.7}$$

where $\mathcal{J}_Y$ denotes the loss of predicting check-in between users and venues, while $\mathcal{J}_{C_U}$ and $\mathcal{J}_{C_V}$ denote the losses of user and venue context predictions respectively. The two values λ_1 and λ_2 are the regularization to control the trade-off among the three losses.

Specifically, $\mathcal{J}_Y$ is the log-loss function which is a special case of cross entropy for softmax input. Formally, it is defined by:

$$\begin{aligned} \mathcal{J}_Y &= \log p(\mathcal{L}|\Theta_e, \Theta_h) \\ &= -\sum_{(u_i,v_j)\in\mathcal{L}^+} \log \hat{y}_{ij} - \sum_{(u_i,v_j)\in\mathcal{L}^-} \log(1-\hat{y}_{ij}) \\ &= -\sum_{(u_i,v_j)\in\mathcal{L}} y_{ij} \log \hat{y}_{ij} + (1-y_{ij}) \log(1-\hat{y}_{ij}) \end{aligned} \tag{8.8}$$

In the above equation, $\mathcal{L}$ represents the collection of labeled check-in pairs of users and venues. $\mathcal{L}$ consists of two subsets $\mathcal{L}^+$ ($\mathcal{L}^+ \subseteq C$) and $\mathcal{L}^-$ ($\mathcal{L}^- \cap C = \varnothing$) corresponding to positive and negative labeled pairs respectively. Θ_e and Θ_h are the parameters used to predict the preference of users and venues. y_{ij} and $\hat{y}_{ij}$ are the actual and prediction of preference of user i and venue j.

The loss functions of user context prediction and venue context prediction, $\mathcal{J}_{C_U}$ and $\mathcal{J}_{C_V}$ are defined by mean square errors:

$$\mathcal{J}_{C_U} = \sum_{u_i} MSE(\hat{u}_i, \mathbf{u}_i) = \sum_{u_i} \|\hat{u}_i - \mathbf{u}_i\|^2 \tag{8.9}$$

where $\hat{u}_i$ is the predicted context vector of user i and $\mathbf{u}_i$ is the actual context vector of user i. We would like to minimize the difference between the two vectors.

$$\mathcal{J}_{C_V} = \sum_{v_j} MSE(\hat{v}_j, \mathbf{v}_j) = \sum_{v_j} \|\hat{v}_j - \mathbf{v}_j\|^2 \tag{8.10}$$

where $\hat{v}_j$ is the predicted context vector of venue j and $\mathbf{v}_j$ is the actual context

vector of venue j.

Model Learning. To learn the parameters Θ_e and Θ_h of the neural network component, we use the optimization technique SGD (stochastic gradient descent) with mini-batch ADAM [41]. The algorithm is the iterative process containing two steps. First of all, we sample the batch of labeled pairs of users and venues from $\mathcal{L}$. Secondly, we optimize the loss functions $\mathcal{J}_Y$, $\mathcal{J}_{C_U}$ and $\mathcal{J}_{C_V}$. We repeat the steps until the loss function converges.

8.2 Experiment

In this section, we describe our experiments on three real world datasets to evaluate our proposed model against relevant baselines. Furthermore, other intensive experiments are also conducted to illustrate the robustness of our model.

8.2.1 Check-in Prediction Task

In this experiment, we evaluate the performance of our model in check-in prediction task. We use three datasets **SG**, **JK** and **NYC**. For each dataset, we sort the check-ins by created time and divide them into the training and testing sets. For the purpose of check-in prediction, we consider the first check-in a user performs on a venue and ignore the subsequent the same user checks into the same venue. The user-venue pairs of these check-ins form the positive data instances. The first 80% of these check-ins forms the training set and the latter 20% forms the testing. We then need to select user-venue pairs for the negative data instances. To keep the positive and negative data instances balanced, we randomly select equal number of user-venue pairs without any check-ins as the negative data instances.

To infer the vector of user/venue context, we apply DeepWalk [66]. The dimension of embedding space of user and venue is 64 (the default setting).

The context graph of users is the social network among them. Specifically, user a connects to user b if a follows b in three datasets. To construct the graph of venues, we assume that venue a and venue b are connected if the physical distance between them is not larger than 100 meter.

Accuracy Measures. To measure the accuracy of prediction results, we use accuracy and F1-score defined by:

$$Accuracy = \frac{\text{Number of Test Instances with Correct Predictions}}{\text{Number of Test Instances}}$$

$$F1 = \frac{2 \times Precision \cdot Recall}{Precision + Recall}$$

where

$$Precision = \frac{\text{Number of Correctly Predicted Check-In Test Instances}}{\text{Number of Predicted Check-In Test Instances}}$$

$$Recall = \frac{\text{Number of Correctly Predicted Check-In Test Instances}}{\text{Number of Check-In Test Instances}}$$

Methods. We evaluate two variants of PACELA method. Other than the full method PACELA, we introduce a variant method PACELA$_v$ that includes only the latent attributes of venues only. We also include the following baseline methods:

- VAN: It is the first model studied neighborhood competition and area attraction [23]. In this model, we use CDF function to model the competition among venues in one area and the size of area is 0.1 degree. The parameters are selected since they generated the best prediction performance [23]. The home location of users are required as input for this model so we estimate the home location of each user by his/her center of the mass of check-ined locations.

- VANF: It is the matrix factorization model to derive the latent attributes of users and venues described in Chapter 6. To use VANF for check-in

Table 8.1: Check-in prediction performance of PACELA and baselines. We boldface the best performance in each dataset.

	Accuracy			F1 score		
	SG	**JK**	**NYC**	**SG**	**JK**	**NYC**
VAN	60.74%	59.93%	55.8%	58.59%	56.66%	58.51%
VANF	75.92%	67.92%	62.12%	68.27%	57.25%	62.45%
PACE	79.3%	66.28%	62.32%	70.84%	57.7%	65.7%
PACELA$_v$	80.1%	70.53%	62%	71.91%	60.49%	66.55%
PACELA	**82.3%**	**72.81%**	**64.59%**	**73.7%**	**61.93%**	**67.92%**

prediction, we learn the matrices U and V from the training data. Unlike the training check-in data used in PACELA, PACELA$_v$ and PACE, we train VANF to learn the actual check-in counts by users on venues. We then use $U_i^T V_j$ to predict for a user-venue pair (i, j). We predict a check-in for the pair if $U_i^T V_j \geq TH$ where TH is a threshold that has been set to 1, as it is the natural threshold to separate the positive from negative instances in our training data. The latent dimension size is set to 10.

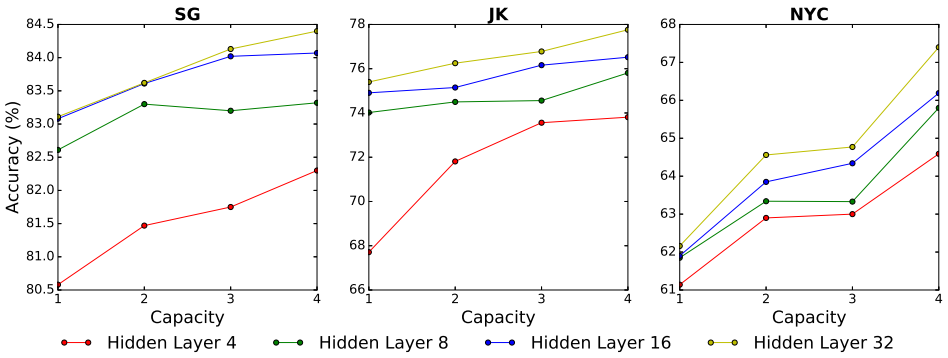
- PACE: PACE method has been proposed in [86] to predict POI visitations. The method learns embedding vectors of users and venues to predict user context, venue context and check-in data in a neural network framework. PACE however does not consider latent attributes of users and venues. As PACELA can be seen as an extension of PACE, we include it for comparison. The multi-layer neural network model of PACE requires two configuration parameters, R capacity and Q number of hidden layers.

Parameter Settings: The default configuration parameters of PACE, PACELA$_v$, and PACELA are capacity R and number of hidden layers Q with default values 4 and 4 respectively. We keep the size of user/venue embedding vector size to 10. The number of latent feature of users and venues in latent attribute modeling is set to 10. For VANF, we set the area size to 0.01, and $\lambda_u = \lambda_v = \lambda_f = 0.01$ since this setting gives the best performance when we use the VANF for check-in prediction task. In model training, we set the batch

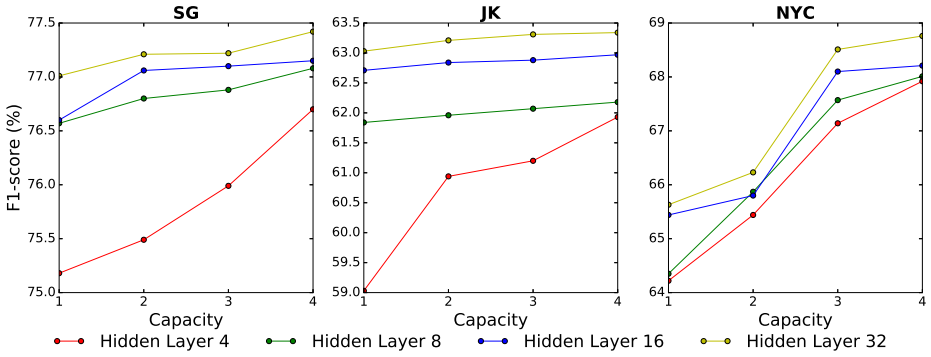
size as 1024, and learning rate as 0.0001.

Results. Table 8.1 provides the accuracy and F1-score of different methods. From the table, we observe that PACELA method outperforms all other methods across the three datasets. For instance, in the **SG** dataset, PACELA has improved 3.7% in accuracy and 4% in F1-score compared with PACE, a state-of-the-art method. We also observe the inclusion of venue latent features also enhances the accuracy of PACE. The PACELA$_v$ method using latent venue features outperforms PACE. This results show that the full PACELA method benefits from latent features from both users and venues.

8.2.2 Parameter Study Experiment



(a) Accuracy



(b) F1-score

Figure 8.2: The prediction performance of PACELA with different values of capacity and number of hidden layers in **SG**, **JK** and **NYC** datasets.

We next evaluate the impact of two configuration parameters R and Q to PACELA method. Recall that R is the capacity which is the length of last output layer of the network while Q is the number of hidden layers. Figures 8.2a

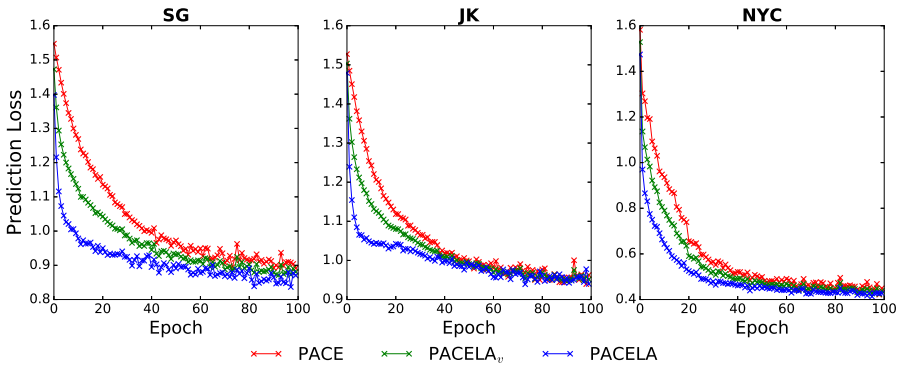


Figure 8.3: The prediction loss of training process in **SG**, **JK** and **NYC** datasets.

and 8.2b show the accuracy and F1-score of PACELA method respectively for different R and Q settings and for the three datasets **SG**, **JK** and **NYC**. In the experiment, we vary R between 1 to 4, and Q between 4 to 32, We seek to determine the performance impact of parameter settings to the methods. The remaining parameters are assigned their default values.

First of all, we observe that higher accuracy can be achieved by PACELA with larger Q values. The improvement however reduces as Q increases to 32. Setting the capacity R higher is also shown to improve accuracy and F1 scores. This can be due to the use of larger neural networks for prediction.

8.2.3 Effectiveness of Latent Attributes of Users and Venues

To gain a deeper understanding of the contribution of user and venue latent attributes, we compare the prediction loss of PACE and PACELA methods through epochs. The faster the convergence of prediction loss, the better the method is.

Experiment Setup. The parameters are set to default values as mention in Section 8.2.1. The number of epochs in this experiment is 100. The three methods that we include in this experiment study are PACE, PACELA$_v$ and PACELA methods.

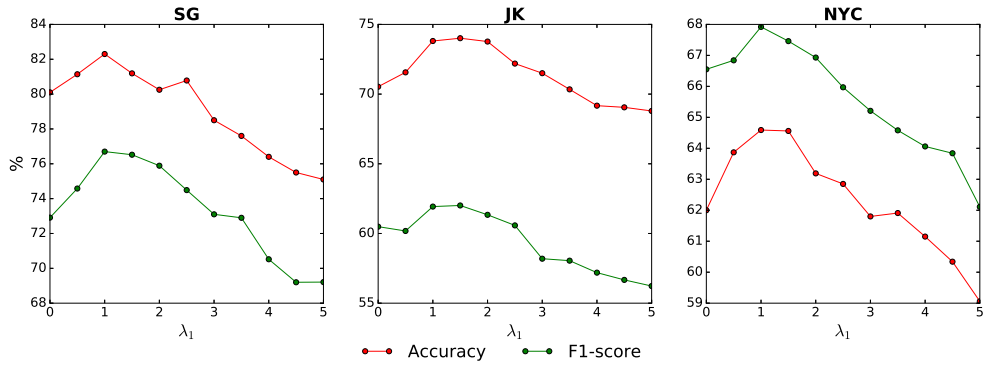
Experiment Results. Figure 8.3 shows the results of the experiment on **SG**, **JK** and **NYC** datasets. As shown in the figure, we observe that.

- When the number of epochs increases, the prediction loss generally decreases. After a certain threshold, the losses become stable and converge to a fixed point.
- The three methods converge to the same stable point in the three datasets. The difference is that the converged values of **SG** and **NYC** datasets are lower than that of **JK** dataset. It could be explained by the fact that **JK** is sparser than **SG** and **NYC**. Our PACELA model requires larger amount of data to achieve the better training loss.
- Finally, the PACELA method extended from PACE by latent attributes of users and venues converges faster than the original model. For instance, at the epoch 10, PACELA method reaches the stable point in the **SG** dataset. The phenomenon clearly happens in the three datasets. It is a clear suggestion that the latent features are useful to enhance the performance of PACELA method.

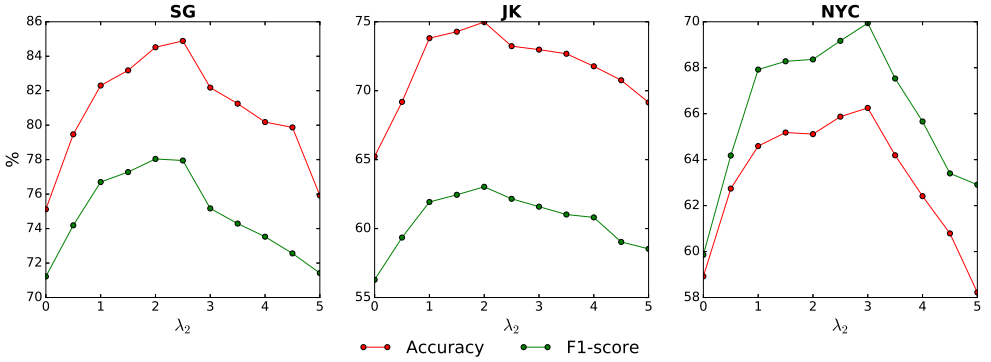
8.2.4 Tuning Regularization

In this experiment, we tune the values of λ_1 and λ_2 in Equation 8.7 to further understand the importance weights of user and venue context to the PACELA model. Recall that λ_1 and λ_2 control the contribution of user context and venue context respectively to the objective function. Setting λ_1 or λ_2 to 0 means that the contribution of user context or venue context is omitted from PACELA method, while increasing λ_1 or λ_2 give higher weights to user or venue context in the PACELA model, respectively.

Setup: We first set $\lambda_1 = 1$ and vary λ_2 from 0 to 5 with step size as 0.5 to evaluate the accuracy of check-in prediction for the PACELA method. Secondly, we repeat the experiment with λ_2 is set to 1 and λ_1 varied between



(a) Tuning λ_1



(b) Tuning λ_2

Figure 8.4: The prediction performance of PACELA under different values of λ_1 and λ_2 in **SG**, **JK** and **NYC** datasets.

0 and 5 with step size 0.5. For other parameters, default values are used (see Section 8.2.1).

Result: Figures 8.4a and 8.4b illustrate the performance of the PACELA method. Our findings include:

- Using user context or venue context improves the accuracy of PACELA. Nevertheless, if we increase the weight of the context too much, it could harm the prediction accuracy. From both figures, we observe that if $\lambda_1 = 0$ or $\lambda_2 = 0$, PACELA yields its lowest accuracy performance. Positive λ_1 or λ_2 values give us prediction accuracy but the improvement declines as these parameters increase. For instance, increasing λ_2 from 0 to 3 improves the accuracy and F1-score of PACELA in NYC dataset, but when λ_2 is greater than 3, the prediction accuracy of PACELA deteriorates.

- Venue context helps to improve PACELA more than user context. Specifically, from the two figures, we observe that the peak performance of PACELA occurs when $\lambda_2 \in [2, 3]$ but $\lambda_1 \in [1, 2]$. The reason for the phenomenon is that we have more information about venues than about users. For example, the number of venues is three times larger than that of users in the **NYC** dataset.

Chapter 9

Conclusion and Future Works

In this chapter, we firstly provide a conclusion for the dissertation. Specifically, we give an overview of our contributions and discuss what we have done through these models. Then, the later part of this chapter proposes some directions for our future research.

9.1 Conclusion

Location-based social networks provide a rich datasets of user movement behaviors. In addition to the historical movement of users, they also contain many valuable information such as the feedbacks of users to venues, the social activity among friends. For this reason, they offer both new research opportunities and challenges for understanding the movement behaviors of users. Motivated by many important applications, our research develop multiple models to capture various effects to model the check-in behavior of users in LBSNs. Our work consists of two parts: (i) modeling various effects without the preference matching between users and venues (ii) modeling various effects considering the preference of users and venues.

The first part includes Chapters 4 and 5. In this part, we consider the neighborhood competition among venues in LBSN.

In Chapter 4, we propose ranking methods using data from location-based

social media. The breakthrough here is to turn check-ins, a kind of visitation data, into competitions between venues and their neighbors. Such an approach is non-intrusive and incurs low overheads [3]. By defining different competitive probability options among venues, and options for combining with check-in ratios, we obtain different PageRank style models. These models have been evaluated on real datasets from Foursquare to determine their differences. We found models based on the competitive probability options behave in very similar way. We have also qualitatively analyzed the results by looking at some interesting cases studies and verify the correctness of our models via the "ground truth" (e.g. Foursquare score). Since it is hard to extend PageRank-based model to capture more effects of check-in behavior of users, there is a need for a more flexible model to handle more effects.

In Chapter 5, we propose the probabilistic VAN model to consider the two factors: neighborhood competition of venues, and area attractiveness in modeling user visitation data. By dividing venues into areas, we could reduce computational cost during learning and inferring processes. Moreover, our learning method is easy to parallelize in order to keep a manageable training time. Finally, the performance of our model is evaluated in three tasks (i.e. home location prediction, venue ranking, and check-in prediction) and its result outperforms the baselines.

The second part includes Chapters 6, 7 and 8. In this part, we model the preference between users and venues by employing matrix factorization based method.

In Chapter 6, we propose a model and its variant that incorporate area attraction, neighborhood competition and social homophily factors. It is enhanced version of VAN model in Chapter 5 since it does not require the exact home location of users and also considers the similarity between the users' preference and the latent characteristics of venues. We evaluate our model in check-in prediction task and show that the proposed model yields better

performance than baselines. Moreover, we also study the performance of our model via different parameter settings to ensure the robustness of our model. Venue-aspect spatial homophily effect is another factor of neighborhood to affect the check-in behavior. However, it is not studied in this model so we need to consider these two features of geographical influence of venues to understand the check-in behavior of users in LBSNs.

In Chapter 7, we model the geographical neighborhood influence of venues to users' check-in behavior by considering spatial homophily and neighborhood competition effects. We proposed the matrix factorization based model to capture the geographical neighborhood influence of venues as well as social homophily effect. In addition to the vector of the intrinsic characteristic, each venue has one more latent feature vector to represent the extrinsic characteristics. The additional vector characterizes the outlooks for each venue. Considering different options to characterize these effects give us the best setting to model such behavior. Moreover, we find out that spatial homophily is not as important as neighborhood competition on predicting the check-in behavior. Finally, social homophily helps our model to improve the accuracy of check-in prediction task.

In Chapter 8, we propose a neural framework named PACELA which embeds the latent attributes of venues and users to improve the preference prediction of users to venues in LBSNs. The user and venue latent attributes are learned by models that exploit behavioral effects in check-ins including those proposed in this dissertation. The framework provides a flexible approach to combine different latent attributes, embeddings of users and venues to predict check-in behaviors.

To summarize, our main contributions in this dissertation are in illustrating the two effects named *neighborhood competition* and *area attraction* and propose several models to study these two effects in order to study the check-in behavior of users in LBSN. Our works can benefit government agencies by

pointing out popular areas so that new roads or subway stations could be built. Moreover, some companies can use our models to find the best place to open their new stores.

9.2 Future Works

To conclude this dissertation, we sketch below some potential directions for future research that can further elaborate our current works.

First, temporal information is an important information to understand the movement of users in LBSNs. For example, people usually travel from home to workspace around 9AM during the weekdays and traverse the opposite way after work hours. Previous works [14, 67, 63] have considered temporal patterns in the modeling of users' check-in behavior. We could therefore extend our proposed models to include temporal patterns. Conceivably, such extended models should be able to predict check-in behavior more accurately.

Secondly, our work assumes that the attractiveness of each area as well as the competitiveness of each venue do not change over time. However, this assumption is overly strict and it needs to be relaxed in order to have a more adaptive model. For example, the service of a particular hotel may be good on the dates without so many customers but it could be worse if the number of customers suddenly increases (e.g. weekends, public holidays). To solve this issue, we require a method that could measure the attractiveness of areas and competitiveness of venues incrementally. The possible technique to handle is online learning [9]. Online learning treats the check-ins of users in LBSNs as a sequential stream and incrementally update the area attractiveness and venue competitiveness. It provides us an incremental measurement of the two scores and also improves the learning time due to the usage of the new coming data. Hence, integrating online learning to modeling the mobility of users in LBSNs brings us many advantages.

Thirdly, area attraction and neighborhood competition are modeled differently in Chapters 5 and 6. Hence, we could treat them similarly but at different levels. In other words, a particular area also competes with others to gain the visitation of users and there is a second competition among its venues to finally attract users. The idea has been used to explain the innovation divergence of different countries [20]. Hence, we intend to propose another model to study neighborhood competition and area attraction under the above assumption. It provides us a chance to research area attraction as a special case of competition.

Finally, with the emergence of multiple social network, users do not restrict themselves to one specific platform. They can use multiple social media platforms for posting and for social activities. Thus, using activities of users across social media platforms can enhance our understanding of users' mobility [50]. For example, if a particular user posts many articles related to food in Twitter, he/she is likely a food lovers. Then, the probability of this kind of users makes check-ins to food-related venues is higher than the one of him/her going to other places. Therefore, enriching our model with external knowledge of users from other social media platforms is also a promising direction.

Bibliography

[1] Amr Ahmed, Liangjie Hong, and Alexander J Smola. Hierarchical geographical modeling of user locations from social media posts. In *WWW*, 2013.

[2] Rachel MacKay Altman. Mixed hidden markov models: an extension of the hidden markov model to the longitudinal data setting. *Journal of the American Statistical Association*, 2007.

[3] Arvind Arasu, Jasmine Novak, Andrew Tomkins, and John Tomlin. Pagerank computation and the structure of the web: Experiments and algorithms. In *WWW*, 2002.

[4] Lars Backstrom, Eric Sun, and Cameron Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In *WWW*, 2010.

[5] Jie Bao, Yu Zheng, and Mohamed F. Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *SIGSPATIAL*, 2012.

[6] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[7] David M Blei and Michael I Jordan. Variational inference for dirichlet process mixtures. *Bayesian analysis*, 2006.

[8] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *JMLR*, 2003.

[9] Léon Bottou. Online algorithms and stochastic approximations. In *Online Learning and Neural Networks*. 1998.

[10] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[11] Jonathan Chang and Eric Sun. Location 3: How users share and respond to location-based data on social networking sites. In *ICWSM*, 2011.

[12] 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*, 2012.

[13] Zhiyuan Cheng, James Caverlee, Kyumin Lee, and Daniel Z Sui. Exploring millions of footprints in location sharing services. In *ICWSM*, 2011.

[14] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: User movement in location-based social networks. In *KDD*, 2011.

[15] Wen-Haw Chong, Bing Tian Dai, and Ee-Peng Lim. Did you expect your users to say this?: Distilling unexpected micro-reviews for venue owners. In *HT*, 2015.

[16] Wen-Haw Chong, Bing Tian Dai, and Ee-Peng Lim. Not all trips are equal: Analyzing foursquare check-ins of trips and city visitors. In *COSN*, 2015.

[17] Wen-Haw Chong, Bing Tian Dai, and Ee-Peng Lim. Prediction of venues in foursquare using flipped topic models. In *ECIR*, 2015.

[18] Justin Cranshaw, Raz Schwartz, Jason I Hong, and Norman Sadeh. The livehoods project: Utilizing social media to understand the dynamics of a city. In *ICWSM*, 2012.

[19] Justin Cranshaw, Eran Toch, Jason Hong, Aniket Kittur, and Norman Sadeh. Bridging the gap between physical location and online social networks. In *UbiComp*, 2010.

[20] Klaus Desmet, Avner Greif, and Stephen L Parente. Spatial competition, innovation and institutions: The industrial revolution and the great divergence. 2017.

[21] Thanh-Nam Doan, Freddy Chong Tat Chua, and Ee-Peng Lim. Mining business competitiveness from user visitation data. In *SBP*, 2015.

[22] Thanh-Nam Doan, Freddy Chong Tat Chua, and Ee-Peng Lim. On neighborhood effects in location-based social networks. In *WI-IAT*, 2015.

[23] Thanh-Nam Doan and Ee-Peng Lim. Attractiveness versus competition: Towards an unified model for user visitation. In *CIKM*, 2016.

[24] Thanh-Nam Doan and Ee-Peng Lim. Modeling check-in behavior with geographical neighborhood influence of venues. In *ADMA*, 2017.

[25] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996.

[26] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[27] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Content-aware point of interest recommendation on location-based social networks. In *AAAI*, 2015.

[28] Huiji Gao, Jiliang Tang, and Huan Liu. gscorr: modeling geo-social correlations for new check-ins on location-based social networks. In *CIKM*, 2012.

[29] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *KDD*, 2007.

[30] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[31] Bo Han, Paul Cook, and Timothy Baldwin. Text-based twitter user geolocation prediction. *Journal of Artificial Intelligence Research*, 2014.

[32] Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alexander J Smola, and Kostas Tsioutsiouliklis. Discovering geographical topics in the twitter stream. In *WWW*, 2012.

[33] Henry Hsu and Peter A Lachenbruch. Paired t test. *Wiley Encyclopedia of Clinical Trials*, 2008.

[34] Bo Hu and Martin Ester. Social topic modeling for point-of-interest recommendation in location-based social networks. In *ICDM*, 2014.

[35] Longke Hu, Aixin Sun, and Yong Liu. Your neighbors affect your ratings: On geographical neighborhood influence to rating prediction. In *SIGIR*, 2014.

[36] D. L Huff. A probabilistic analysis of shopping center trade areas. *Land Economics*, 1963.

[37] Zongcheng Ji, Aixin Sun, Gao Cong, and Jialong Han. Joint recognition and linking of fine-grained locations from tweets. In *WWW*, 2016.

[38] Michael I Jordan et al. Why the logistic function? a tutorial discussion on probabilities and neural networks, 1995.

[39] Kenneth Joseph, Chun How Tan, and Kathleen M Carley. Beyond local, categories and friends: clustering foursquare users with latent topics. In *UbiComp*, 2012.

[40] Dmytro Karamshuk, Anastasios Noulas, Salvatore Scellato, Vincenzo Nicosia, and Cecilia Mascolo. Geo-spotting: mining online location-based services for optimal retail store placement. In *KDD*, 2013.

[41] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[42] Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *KDD*, 2008.

[43] Yehuda Koren, Robert Bell, Chris Volinsky, et al. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[44] John Krumm. Inference attacks on location tracks. In *PERVASIVE*, 2007.

[45] John Krumm and Eric Horvitz. Predestination: Inferring destinations from partial trajectories. In *UbiComp*. 2006.

[46] Amy N Langville and Carl D Meyer. Deeper inside pagerank. *Internet Mathematics*, 1(3):335–380, 2004.

[47] Géraud Le Falher, Aristides Gionis, and Michael Mathioudakis. Where is the soho of rome? measures and algorithms for finding similar neighborhoods in cities. In *ICWSM*, 2015.

[48] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 2015.

[49] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2001.

[50] Roy Ka-Wei Lee, Tuan-Anh Hoang, and Ee-Peng Lim. On analyzing user topic-specific platform preferences across multiple social media sites. In *WWW*, 2017.

[51] Huayu Li, Yong Ge, and Hengshu Zhu. Point-of-interest recommendations: Learning potential check-ins from friends. In *KDD*, 2016.

[52] Huayu Li, Richang Hong, Shiai Zhu, and Yong Ge. Point-of-interest recommender systems: A separate-space perspective. In *ICDM*, 2015.

[53] Huayu Li, Hong Richang, Wu Zhiang, and Yong Ge. A spatial-temporal probabilistic matrix factorization model for point-of-interest recommendation. In *SDM*, 2016.

[54] Rui Li, Shengjie Wang, Hongbo Deng, Rui Wang, and Kevin Chen-Chuan Chang. Towards social user profiling: unified and discriminative influence model for inferring home locations. In *KDD*, 2012.

[55] Yanhua Li, Moritz Steiner, Limin Wang, Zhi-Li Zhang, and Jie Bao. Exploring venue popularity in foursquare. In *INFOCOM*, pages 3357–3362, 2013.

[56] Zhenhui Li, Bolin Ding, Jiawei Han, Roland Kays, and Peter Nye. Mining periodic behaviors for moving objects. In *KDD*, 2010.

[57] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. Modeling user exposure in recommendation. In *WWW*, 2016.

[58] Janne Lindqvist, Justin Cranshaw, Jason Wiese, Jason Hong, and John Zimmerman. I'm the mayor of my house: examining why people use foursquare-a social-driven location sharing application. In *CHI*, 2011.

[59] Bin Liu, Yanjie Fu, Zijun Yao, and Hui Xiong. Learning geographical preferences for point-of-interest recommendation. In *KDD*, 2013.

[60] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. Exploiting geographical neighborhood characteristics for location recommendation. In *CIKM*, 2014.

[61] Samantha Lundrigan and David Canter. Spatial patterns of serial murder: An analysis of disposal site location choice. *Behavioral sciences & the law*, 19(4), 2001.

[62] Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, 2007.

[63] Anastasios Noulas, Salvatore Scellato, Neal Lathia, and Cecilia Mascolo. Mining user mobility features for next place prediction in location-based services. In *ICDM*, 2012.

[64] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[65] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 2011.

[66] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *KDD*, 2014.

[67] Tatiana Pontes, Marisa A. Vasconcelos, Jussara M. Almeida, Ponnurangam Kumaraguru, and Virgilio Almeida. We know where you live: privacy characterization of foursquare behavior. In *UbiComp*, 2012.

[68] Daniel Preoţiuc-Pietro, Justin Cranshaw, and Tae Yano. Exploring venue-based city-to-city similarity measures. In *UrbComp*, 2013.

[69] Yan Qu and Jun Zhang. Trade area analysis using user generated mobile location data. In *WWW*, 2013.

[70] John R Quinlan et al. Learning with continuous classes. In *5th Australian joint conference on artificial intelligence*, volume 92, pages 343–348. Singapore, 1992.

[71] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, 2009.

[72] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. A metric for distributions with applications to image databases. In *ICCV*, 1998.

[73] Salvatore Scellato, Anastasios Noulas, Renaud Lambiotte, and Cecilia Mascolo. Socio-spatial properties of online location-based social networks. *ICWSM*, 2011.

[74] Salvatore Scellato, Anastasios Noulas, and Cecilia Mascolo. Exploiting place features in link prediction on location-based social networks. In *KDD*, 2011.

[75] Mikkel N Schmidt, Ole Winther, and Lars Kai Hansen. Bayesian non-negative matrix factorization. In *Independent Component Analysis and Signal Separation*, pages 540–547. Springer, 2009.

[76] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits of predictability in human mobility. *Science*, 327(5968), 2010.

[77] Dan Tasse and Jason I Hong. Using social media data to understand cities. In *Workshop on Big Data and Urban Informatics*, 2014.

[78] Dan Tasse, Alex Sciuto, and Jason I Hong. Our house, in the middle of our tweets. In *ICWSM*, 2016.

[79] Eran Toch, Justin Cranshaw, Paul Hankes-Drielsma, Jay Springfield, Patrick Gage Kelley, Lorrie Cranor, Jason Hong, and Norman Sadeh.

Locaccino: a privacy-centric location sharing application. In *UbiComp*, 2010.

[80] Hanghang Tong, Christos Faloutsos, and Jia Y Pan. Fast random walk with restart and its applications. In *ICDM*, 2006.

[81] Janice Y Tsai, Patrick Kelley, Paul Drielsma, Lorrie Faith Cranor, Jason Hong, and Norman Sadeh. Who's viewed you?: the impact of feedback in a mobile location-sharing application. In *CHI*, 2009.

[82] Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In *KDD*, 2011.

[83] Lilian Weng, Alessandro Flammini, Alessandro Vespignani, and Filippo Menczer. Competition among memes in a world with limited attention. *Scientific Reports*, 2012.

[84] Jason Wiese, Patrick Gage Kelley, Lorrie Faith Cranor, Laura Dabbish, Jason I Hong, and John Zimmerman. Are you close with me? are you nearby?: investigating social groups, closeness, and willingness to share. In *UbiComp*, 2011.

[85] Xiao-Yong Yan, Wen-Xu Wang, Zi-You Gao, and Ying-Cheng Lai. Universal model of individual and population mobility on diverse spatial scales. *Nature Communications*, 8(1):1639, 2017.

[86] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. Bridging collaborative filtering and semi-supervised learning: A neural approach for poi recommendation. In *KDD*, 2017.

[87] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhu Wang. A sentiment-enhanced personalized location recommendation system. In *HT*, 2013.

[88] Jihang Ye, Zhe Zhu, and Hong Cheng. What's your next move: User activity prediction in location-based social networks. In *SDM*, 2013.

[89] Mao Ye, Krzysztof Janowicz, Christoph Mülligann, and Wang-Chien Lee. What you are is when you are: the temporal dimension of feature types in location-based social networks. In *SIGSPATIAL*, 2011.

[90] Mao Ye, Peifeng Yin, and Wang-Chien Lee. Location recommendation for location-based social networks. In *SIGSPATIAL*, 2010.

[91] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, 2011.

[92] Yang Ye, Yu Zheng, Yukun Chen, Jianhua Feng, and Xing Xie. Mining individual life pattern based on location history. In *MDM*, 2009.

[93] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Who, where, when and what: discover spatio-temporal topics for twitter users. In *KDD*, 2013.

[94] Yingjie Zhang, Beibei Li, and Jason Hong. Understanding user economic behavior in the city using large-scale geotagged and crowdsourced data. In *WWW*, 2016.

[95] Tong Zhao, Julian McAuley, and Irwin King. Leveraging social connections to improve personalized ranking for collaborative filtering. In *CIKM*, 2014.

[96] Yu Zheng, Lizhu Zhang, Zhengxin Ma, Xing Xie, and Wei-Ying Ma. Recommending friends and locations based on individual location history. *TWEB*, 5(1):5, 2011.