NEARBY PLACES:

ON LOCATION-BASED PRUNING FOR

POINT-OF-INTEREST RECOMMENDATION

JORDAN SILVA

NEARBY PLACES: ON LOCATION-BASED PRUNING FOR POINT-OF-INTEREST RECOMMENDATION

Dissertation presented to the Graduate Program in Computer Science of the Universidade Federal de Minas Gerais Departamento de Ciência da Computação in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Rodrygo Luis Teodoro Santos
Co-Advisor: Nivio Ziviani

Belo Horizonte
September 2017

Silva, Jordan

S586n Nearby Places: On Location-based Pruning for Point-of-interest Recommendation / Jordan Silva. — Belo Horizonte, 2017
xxiii, 76 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais Departamento de Ciência da Computação
Orientador: Rodrygo Luis Teodoro Santos
Coorientador: Nivio Ziviani

1. Computação — Teses. 2. Recuperação da informação. 3. Sistemas de recomendação. 4. Redes sociais on-line. I. Orientador. II. Coorientador. III. Título.

CDU 519.6*73(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Nearby places: on location-based pruning for point-of-interest recommendation

JORDAN SILVA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

RLSantos

PROF. RODRYGO LUIS TEODORO SANTOS - Orientador
Departamento de Ciência da Computação - UFMG

Nivio Ziviani

PROF. NIVIO ZIVIANI - Coorientador
Departamento de Ciência da Computação - UFMG

PROF. ANISIO MENDES LACERDA
Departamento de Computação - CEFET

PROF. PEDRO OLMO STANCIOLI VAZ DE MELO
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 27 de setembro de 2017.

Acknowledgments

My next words are for thank to all whom I met and helped me go through this amazing journey, my master's.

First, I would like to thank my advisor Rodrygo Santos, who believed and supported me over these three years. Despite my ups and downs, Rodrygo encouraged and motivated me throughout the work. Thank you. Also, I would like to thank my co-advisor Nivio Ziviani, from whom I learned a lot and also have great respect for. I have to thank the entire team of UFMG that collaborated to make this process be as easy as possible: Linda, Sônia, Sheila.

This work would not be possible without my friends. My beloved friends, good listeners, good advisers, great people that I had good talks with. Rafael Glater, thanks for being crazy enough to start this journey together. Daniel Hasan, thanks for introducing me to Rodrygo and making it all possible. Many thanks to my peers who shared great moments together: Alberto Ueda, Arthur Câmara, Bráulio Veloso, Bruno Laporais, Felipe Moraes, Gustavo Penha, João Matheus, Lucas Chaves, Marlon Dias, Raul Sanchez, Vitor Mangaravite, Wellington Dores, Raphael Campos, Ana Paula Gomes, Harlley Lima. Also, many thanks to the old school from LATIN which I had the pleasure of meeting: Sabir Ribas, Anisio Lacerda, Aécio Santos, Cristiano Carvalho, Thales Costa.

Last, but not least, a huge thanks to my family who supported me through everything and without them I would not have done this dissertation. Thank my parents Rita and Mardoqueu, and my siblings, Milene and Gibran. Also, thank you Gabriela, my love, for putting up with me when I was very stressed and being very annoying. I love you.

"The master has failed more times than the beginner has even tried."

(Stephen McCranie)

Resumo

A recomendação de pontos de interesse (POIs, na sigla em inglês) em redes sociais baseadas em localização tem sido objeto de intensa pesquisa, como mecanismo para auxiliar usuários na descoberta de novos lugares para visitação em uma cidade. Tipicamente, todos os POIs na cidade alvo são considerados como candidatos para recomendação, independentemente de quão distantes estão localizados do usuário alvo. Em contraste, pesquisas recentes mostram que os usuários tendem a visitar POIs localizados em sua vizinhança a maior parte do tempo. A fim de explorar esse padrão de localidade, investigamos o impacto de um mecanismo de poda baseado em localização para a recomendação de POIs. Em particular, mostramos que ganhos de eficiência podem ser obtidos durante as fases de aprendizado e de predição de recomendações sem perdas significativas de eficácia. Por meio de uma análise segmentada, demonstramos ainda o compromisso entre eficácia e eficiência para diferentes níveis de esparsidade dos perfis de visitação de usuários e POIs, bem como para vizinhanças de várias densidades.

Abstract

Point-of-interest (POI) recommendation in location-based social networks has been widely researched as a mechanism to help users discover new places to visit in a city. Typically, all POIs in the target city are considered as candidates for recommendation, regardless of how far off they are from the target user. In contrast, related research has shown that users tend to visit POIs in their vicinity most of the time. To exploit such a locality pattern, we investigate the impact of location-based pruning on POI recommendation. In particular, we show that efficiency improvements can be attained both at learning as well as at prediction time without significant losses in effectiveness. Through a breakdown analysis, we further demonstrate the effectiveness-efficiency trade-offs incurred for different levels of user and POI visitation sparsity as well as for neighborhoods of various densities.

List of Figures

31 The collector's architecture. 16

32 The Foursquare shared check-in on Twitter. 17

33 Check-ins coverage at the target cities. 18

34 Distribution check-ins by distance in the New York City. 22

41 The number of candidate POIs per recommendation task. (a) The traditional task considers every place in the city as a possible candidate. Each point denotes a POI. The colors express different POI categories. (b) The Nearby task explores only the users' neighborhood, from the users' location. The major circles represent the size of the neighborhood and the minor circles are the candidate POIs; the marker on the circle's center is the user's location and other the relevant POI. The colors mean different check-ins. . 28

51 Test collection divided into 8-time windows. 5-boxes compose a time-window, and each box is a one-month check-in. The first three months inside the brackets, in the color blue, are the data for training. The next two boxes are validation (yellow) and test data (green), respectively. Above the time-window has a #<number> that represents the time-window tag. 33

61 Relevant item frequency normalized by a set of candidate POIs. Higher values represent that relevant venue is most popular among the neighborhood. 41

62 Difference between the reciprocal rank of WBPRMF and BPRMF. ρ is the popularity of the relevant POI normalized by its set of candidates. The higher value, the more popular the relevant POI is in its neighborhood. . . 43

63 Accuracy of methods for users with different numbers of check-ins. Memory-based methods are represented by a line with no marker; Model-based methods are a dotted line; and Non-personalized methods are indicated by a line with markers. 44

64 Win-loss comparison between approaches when the user is cold-start. The red color indicates that the row-algorithm scores higher than column-algorithm, otherwise blue ones indicate a lower score. In Figure 64a, the value in each cell is the row-algorithm percentage of winning test cases over the column-algorithm. The cell value in Figure 64b is the difference between the MRRs of the row and columns algorithms. 45
65 The standard deviation of the set of candidates popularity in system cold-start scenario. 46
66 Distribution of the number of candidates per test case (only non-cold-start test cases). 48
67 Distribution of the relevant item popularity with non-cold-start test cases. Item popularity is normalized and represent the relative popularity of the POI in the set of candidates. 48
68 Relevant item frequency normalized by a set of candidates. Only non-cold-start test cases. 49
69 Distribution of the rank size using user's location in a range of 500 meters and 1 kilometer from the relevant POI. 50
610 The accuracy of the algorithms using 500 meters and 1 kilometer in the radius size of the neighborhood. 51
611 The accuracy of the algorithms using the neighborhood densities. A radius $r = 500$ was used to build the set of candidate POIs. 52
612 The accuracy of the algorithms by the neighborhood score in item cold-start scenario. The neighborhood score is the sum of the popularity of the k closest POIs of the target POI. $k = 100$. 52
613 Time comparison between the Rank-GeoFM and Nearby Rank-GeoFM over 1000 iterations. Axis-x shows the time elapsed in seconds to complete the iteration. 55

List of Tables

31 The contextual features available in Nearby POI collection 19
32 Nearby POI Collection 22
33 List of POI categories 22

51 List of notations 32
52 Datasets 32

61 A performance comparison pruning the test cases with different neighborhood ranges in the New York City collection. # is the relative position of the method by the MRR metric. Neighborhood and user's location divides into 500 meters, 1 kilometer and traditional which does not use restriction of distance. 38
62 A performance comparison of different algorithms for the Nearby POI Recommendation task in the New York City collection using 500 meters as neighborhood range. Results are divided in five categories, i.e., three cold-start scenarios, a non cold-start, and the overall result (see Appendix C2 and C3 for the other collections). 40
63 A effectiveness comparison pruning the training process of the Rank-GeoFM. Nearby Rank-GeoFM is the modified algorithm to pruned the training considering only the nearby POIs as candidates. MRR is measured in two different neighborhood and user's locations ranges: 1 kilometer and 500 meters. 54
64 A comparison of iteration time by pruning training data of the Rank-GeoFM in New York City collection. 54

A1 The list of fields on a tweet 63
A2 The list of climate features 63
A3 The full list of available features in Nearby POI collection 64

A4 Most commonly collections for POI Recommendation. The following marks indicate that the collection ✔ possesses such information. 65

C1 The Nearby POI recommendation task in different collections. 69
C2 The effectiveness of recommenders for Nearby POI Recommendation task in the Singapore collection. 69
C3 The effectiveness of recommenders for Nearby POI Recommendation task in the São Paulo collection. 70
C4 A performance comparison using items with different numbers of check-ins in the Singapore collection. 70
C5 A performance comparison using items with different numbers of check-ins in the São Paulo collection. 70

List of Algorithms

1 Nearby Rank-GeoFM 68

Contents

Acknowledgments **vii**

Resumo **xi**

Abstract **xiii**

List of Figures **xv**

List of Tables **xvii**

1 Introduction **1**

1.1 Dissertation Statement 2

1.2 Contributions 3

1.3 Organization 4

2 Related Work **5**

2.1 Recommender Systems 5

2.2 POI Recommendation 7

2.2.1 POI Recommendation Techniques 9

2.3 Static Index Pruning in Web Search Engines 11

2.4 Summary 12

3 Nearby Point-of-Interest Collection **15**

3.1 Check-ins Collection 15

3.1.1 Twitter 15

3.1.2 Swarm 17

3.2 Context-Aware Enrichment 19

3.2.1 Temporal Context 19

3.2.2 Climatic Context 20

3.2.3 Social Context 20

3.2.4 Geographical Context 21
3.3 Dataset Characterization 22
3.3.1 Considerations 23
3.4 Summary 23

4 Nearby POI Recommendation 25
4.1 User's Location 26
4.2 User's Neighborhood 27
4.3 Overview 29
4.4 Summary 30

5 Experimental Setup 31
5.1 Problem Definition 31
5.2 Dataset 32
5.3 Evaluation Methodology 33
5.4 POI Recommenders 34
5.4.1 Non-Personalized Recommender Systems 34
5.4.2 Personalized Recommender Systems 35
5.4.3 Parameter Settings 36
5.5 Summary 36

6 Experimental Evaluation 37
6.1 Pruned Test 37
6.2 Breakdown Analyses 40
6.2.1 Check-in Sparsity 40
6.2.2 Neighborhood Density 49
6.3 Pruned Training 53
6.4 Summary 55

7 Conclusions and Future Work 59
7.1 Summary of Contributions 59
7.2 Directions for Future Research 61

Appendix A Nearby POI Collection 63

Appendix B Nearby Rank-GeoFM 67
B.1 Complexity Analysis 67

C Additional Results 69

Bibliography **71**

Chapter 1

Introduction

The rise of location-based social networks (LBSNs), such as Yelp[1] and Foursquare[2] has attracted much attention from industry and academia. For instance, Foursquare contains more than 50 million active users around the world, and by July 2017 had surpassed more than 12 billion check-ins[2]. People have been extensively using LBSNs to achieve a better quality of life, especially in smart cities where every person and device share information on the Web. In particular, users connect with friends, explore their current region and eventually check-in at a *point-of-interest* (POI) (e.g., eatery, entertainment, professional, public service venues) and leave reviews (sharing their experience). A major role of LBSNs consists in learning user's preferences to recommend personalized POIs that would improve their life while exploring a city. POI recommendation not only helps users to visit new places, but also helps the LBSNs to increase revenue by using intelligent location services and attracting potential visitors to a POI [Yu and Chen, 2015].

The large volume of social data generated through LBSNs opens up numerous challenges for providing a high-quality recommender system for citizens. While developing new algorithms for POI recommendation, many researchers have struggled with the high sparsity of user information, the temporal patterns of user activities, and the availability of data from multiple sources [Bao et al., 2015; Ye et al., 2013; Zhao et al., 2016]. A key challenge in this domain is how to assess the effectiveness of a POI recommender given the user's context, such as his or her current location, time, companion, and environmental conditions. Indeed, most previous works on POI recommendation overlook the contextual information, providing for an arguably unrealistic evaluation scenario [Yuan et al., 2013].

[1] `http://www.yelp.com/about`
[2] `http://foursquare.com/about`

Typically, in these studies, all POIs in the target city are considered as candidates for recommendation, regardless of how far off they are from the target user. On the other hand, related research has shown that users usually tend to visit POIs in their vicinity [Ye et al., 2011; Noulas et al., 2011; Liu et al., 2014, 2013]. Considering all POIs as plausible candidates may lead to increase potential noise in the recommendation and a high computational cost to predict a score for POIs outside the user's geographic range. Moreover, ignoring the user's location in the recommendation may trivially favor geography-aware POI recommenders when geography should be a fundamental user constraint.

In this dissertation, we investigate the geographical context of the user on POI recommendation. Unlike previous works that consider every POI as a candidate, we propose a task which leverages the user's location to prune out-ranged POIs from the recommendation. Akin to static pruning in search engines, the Nearby POI recommendation performs a location-based pruning strategy to restrict the recommendations to POIs in the user's vicinity. To demonstrate the impact of the location-based pruning on POI recommendation, we evaluate eight state-of-the-art recommender methods on a new test collection for context-aware nearby POI recommendation.

In particular, the analyses demonstrate that location-based pruning improves the efficiency of the methods at learning and prediction time without significant losses in effectiveness. Preliminary results show that Rank-GeoFM [Li et al., 2015], the current state-of-the-art for POI recommendation, benefits from the location-based pruning, reducing training time by 22%. Moreover, a breakdown analysis further demonstrates the impact on the effectiveness of all tested POI recommenders when pruning is performed at prediction time for users and POIs for different levels of check-in data sparsity as well as for neighborhoods of various densities.

1.1 Dissertation Statement

The statement of this dissertation is that the geography should be a fundamental user constraint to POI recommendation. Besides providing for a more realistic assessment of POI recommendation, we hypothesize that a location-based pruning of the available POIs can improve the efficiency of existing recommenders without significantly degrading their effectiveness. In addition to better reflecting users check-in behavior, factoring out geography during evaluation enables a clearer assessment of the role of contextual factors other than geography on a user's decision to check in at a POI. This statement raises the following research questions, which will be answered in the upcoming chapters:

- **RQ1:** How does location-based pruning impact POI recommendation effectiveness and efficiency?

- **RQ2:** How do different levels of data sparsity affect the effectiveness of POI recommendation methods?

1.2 Contributions

This dissertation leads to the following contributions:

1. A large-scale test collection and evaluation methodology for context-aware nearby POI recommendation.

   In this dissertation, we collect users' check-ins spread over one year from the entire world. We describe the developed architecture for collecting and processing these check-ins. In contrast to existing collections, we propose a geography-aware test collection which includes several geography-constrained test cases, enriched with temporal, social and weather contexts.

2. An effectiveness-efficiency analysis regarding the impact of location-based pruning on POI recommendation.

   As known from the literature, users tend to visit nearby POIs most of the time [Liao et al., 2016; Ye et al., 2011; Liu et al., 2017]. To exploit such pattern, we introduce the Nearby POI recommendation task which uses a location-based pruning for POI recommendation. Also, we present preliminary results for a variation of Rank-GeoFM (Nearby Rank-GeoFM) that shows efficiency improvements both at training and at prediction time without losses in effectiveness.

3. A breakdown analysis comparing the performance of eight POI recommenders through different levels of sparsity and neighborhood densities.

   We compare several state-of-the-art POI recommenders for the Nearby POI recommendation task, pruning the test data. Through a breakdown analysis, we evaluate the performance of the methods for different sparsity conditions (e.g., user and item cold-start, system cold-start) as well as the impact of having the geographical context built into the task.

To the best of our knowledge, this is the first work that proposes a static pruning strategy and evaluates the impact of the location-based pruning for POI recommendation.

1.3 Organization

The remainder of this dissertation is structured as follows:

Chapter 2 gives an overview of recommender systems. The unique challenges of POI recommendation compared with tradition recommendation is also addressed in this chapter. We also discuss existing approaches and relevant results in order to motivate our research questions.

Chapter 3 describes the construction of the Nearby POI collection. We discuss the challenges involved in collecting this data, which is not directly made available by LBSNs for safety reasons. Lastly, we present an analysis and characterization of the produced collection as well as its limitations.

Chapter 4 presents the Nearby POI Recommendation task which restricts the user's POI exploration to his or her vicinity. Because people tend to visit nearby POIs most of the time, we introduce a location-based pruning strategy to exploit such locality pattern. In addition, this chapter describes the main characteristics of the task in comparison to the traditional POI recommendation task.

Chapter 5 details the experimental framework employed to conduct our experiments. Also, we review the problem definition as well as the statistics of the test cases produced. The POI recommenders used in our experiments are presented in this chapter.

Chapter 6 evaluates the performance of eight POI recommenders for the Nearby POI Recommendation task. In particular, the chapter presents an analysis of the impact of location-based pruning on POI recommendation and conduct a breakdown analysis comparing the effectiveness-efficiency of the multiple recommenders for different user-POI sparsity conditions.

Chapter 7 concludes this dissertation by presenting critical aspects and limitations. Furthermore, we summarize the main contributions and directions for future work.

Chapter 2

Related Work

This chapter introduces the theoretical background and the relevant concepts for this dissertation. Section 2.1 gives an overview of the Recommender System describing the task. Later, it explains the recommendation process and its different approaches. Section 2.2 presents the POI Recommendation task and discusses several POI recommender systems divided into context-agnostic and context-aware approaches. Section 2.3 reviews the static pruning in search engines which we employ a similar approach for POI recommendation. Finally, based on the analysis of related work we verify the research gap and determine our research question.

2.1 Recommender Systems

Recommender Systems (RSs) have been studied and developed since the 1990s. At the beginning, recommender systems were described as systems for which "*people provide recommendations as inputs which the system then aggregates and directs to appropriate recipients*" [Resnick and Varian, 1997]. Nowadays the term has a broader connotation, described by Burke [2002] as "*any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options.*"

Since last decade, recommender systems are changing the manner people find products, information, and even a relationship. RSs are popular systems that leverage various techniques to suggest items that users are likely to be interested [Yang et al., 2013]. These suggestions relate to several decision-making processes, such as which movies to watch, what music to listen or where to go [Ricci et al., 2010]. In addition, studies are being carried out to analyzing patterns of users' behaviors to find out what they might like from a collection of things they have never seen before, hence, get a better recommendation.

A recommendation divides into two categories, personalized and non-personalized recommender systems. As a rule, recommender systems focus on a particular type of item, using personalized recommendation algorithms to boost its recommendations. These algorithms give personalized recommendations by user or user group based on their behaviors and uniqueness. In contrast, the non-personalized recommendations aim to everyone, for instance, recommending a list of most popular items.

Recommendation Systems have been evolving over the last years, developing several techniques to predict the most likely item for the user. According to the literature [Burke, 2002], these techniques divide into five different types of recommendation as follows:

Collaborative Filtering

The Collaborative Filtering (CF) is considered the most popular and adopted recommendation technique in RSs. This approach recommends items that other users with similar tastes liked in the past. The similarity between users is calculated based on their rated items before, i.e., users with similar taste rate items similarly [Ricci et al., 2010].

Content-Based

Content-based RSs center on recommending items that are most similar to the ones that the user liked in the past. Thus, the primary process consists of matching up the attributes of a user profile in which preferences and interests are stored, with the attributes of a content item, to recommend to the user new interesting items [Ricci et al., 2010]. For instance, if a user has liked an Italian restaurant, the system can learn to recommend others Italians restaurants or with similar attributes.

Knowledge-Based

Knowledge-Based RSs takes into consideration the knowledge about the items, such features, and build the user's preferences weighting most important features against others. This recommendation algorithm suggests items based on user's feedbacks, e.g., system navigation, searching, specific features filtering. For instance, the eBay[1] implements this approach recommending a set of items learned by the user's features

[1] `http://www.ebay.com`

preferences through the filtering product features, colors, price, etc. Burke [2000] describes knowledge-based recommender systems are strongly complementary to other types of recommender systems, once its recommendations do not depend on a base of user ratings.

Demographic-Based

A Demographic RSs provide the recommendation based on demographic information of the user. In this RS, socio-demographic attributes are used, such as gender, age, education, country. Therefore, grouping user preferences into demographic clusters can improve the recommendation for different demographic niches [Ricci et al., 2010].

Hybrid

Hybrid RSs combine several techniques mentioned above to recommend items. It allows the RS build a more robust system which eliminates the disadvantages of one system with the advantages of another one [Gorakala and Usuelli, 2015]. For instance, content-based systems can recommend new items that have no ratings, while collaborative filtering systems are better with more items feedback.

2.2 POI Recommendation

POI recommendation is a branch of recommendation systems, which indicates that this task can borrow many ideas from conventional recommender systems (e.g., movie recommendation) [Zhao et al., 2016]. For instance, we can apply conventional techniques for POI recommendation such as collaborative filtering methods. POI recommendation systems seek to learn the users' preferences on locations and provide to users a new set of POIs that they are likely to be interested in but has never visited. This task not only encourages users to visit new places but also helps the POI owners to increase their revenues by finding and attracting potential visitors [Liu et al., 2013]. However, differing from traditional recommender systems, POI recommendations face new challenges as follows:

Physical constraints. The user's location plays a vital role for POI recommendation in LBSNs. A user's current location can be represented on different levels of granularity, e.g., a state, city or neighborhood. We should use a fine granularity regarding the kind of recommendation we intend to provide, like using the city as user's location to suggest touristic places to check-in [Bao et al., 2015]. Another concern about the physical constraint is that user's mobility preference is affected by geographical distance. Geographical influence is the most important characteristic that distinguishes POI recommender systems from traditional recommender systems and heavily effects user's visitation behaviors [Yu and Chen, 2015]. Also, the distance property of POIs implies that users are more likely to visit nearby POIs than distant ones [Ye et al., 2011; Noulas et al., 2011; Liu et al., 2014, 2013; Bao et al., 2015; Liu et al., 2017].

Data scarcity problem. POI recommendations suffer from much worse data scarcity problem than other recommendation problems. In traditional recommender systems, the user generally expressed their preferences by explicitly feedbacks (e.g., rating a movie or a product) whereas in the POI recommender systems the user's preferences are reflected by the frequency of check-ins (i.e., implicit feedbacks). A problem of user's preferences being reflected through implicit feedbacks is that a check-in does not mean that the user liked the visited place. In addition, the sparsity of user-location check-in is dramatically higher than of user-item rating matrix, for instance, the sparsity of Netflix data for movie recommendation is 98.8% [Liu et al., 2017], while the collection used in this dissertation is 99.9%.

Heterogeneous information. LBSNs consist of different kinds of information. In addition to geographical information, also consist of venue descriptions, users' social relation and media information (e.g., tips, reviews, likes). The heterogeneous information might be used to improve and construct different kinds of POI recommendation systems. For example, a location shared by two users could be evidence of similarity, social-ties, or it may simply be a very popular place. Moreover, the users' visiting preferences might be affected by their social ties [Liu et al., 2017]. In addition, the users' preference is time-dependent. Users' check-ins pattern may behave differently with respect to time, for instance, a user often checks in parks in the early morning while other user likes bars and checks in late night. The same befalls POIs because different places have different opening hours. Thus their check-ins patterns over time will be different [Cheng et al., 2011].

In the last five years, the POI recommender systems have been receiving a lot of attention from academia and industry. As a result, several POI recommender systems have emerged using different approaches as well as leveraging context information to improve the POI recommendation. Despite that, in general, POI recommender systems have the following structure, a set of n users $\mathcal{U} = \{u_1, \ldots, u_n\}$, and a set of m POIs $\mathcal{P} = \{l_1, \ldots, l_m\}$. Each user u_j has a set of visited POIs $\mathcal{P}_{uj} \subseteq \mathcal{P}$. The check-ins information are often transformed to a user-location check-in frequency matrix C, which each user-location check-in is c_{up}. The users' check-in frequency reflect their preferences through the places visited. However, as aforementioned, matrix C is very sparse due to the small amount of POIs that the users usually check-in.

2.2.1 POI Recommendation Techniques

Due to the lot attention that POI recommendations have been receiving in the last years, an extensive number of studies around the POI recommenders have been developed. Therefore, for a better understanding, we divide these recommenders into context-agnostic and context-aware POI recommendation.

Context-Agnostic POI Recommendation

Like traditional recommender systems, the context-agnostic POI recommenders only take user-item matrix to produce POI recommendations. Therefore, adopting the POIs as the items and the check-ins as ratings, several traditional recommender systems can be used to learn the users' preferences and produce POI recommendations. For instance, traditional techniques like memory-based approaches (e.g., user-based and item-based collaborative-filtering) were employed as POI recommendation algorithms by Ye et al. [2011]. In a nutshell, the user-based collaborative filtering (CF) algorithm assumes that similar users have similar tastes while the item-based CF assumes that users are interested in similar items [Breese et al., 1998; Sarwar et al., 2001]. Thus, user-based CF suggests POIs based on the taste of most similar neighbors and the item-based CF suggests based on similar POIs that the user have visited.

In addition, model-based CF approaches such as matrix factorization (MF) algorithms can be employed for POI recommendation. In contrast to memory-based CF approach, this approach uses different kinds of algorithms to build a model and predicts the user's ratings. MF decomposes the check-in matrix $\mathbf{C} \in \mathbb{R}^{M \times N}$ into a user matrix $\mathbf{U} \in \mathbb{R}^{M \times K}$ and POI matrix $\mathbf{P} \in \mathbb{R}^{N \times K}$. M is the number of the users, N is the number of POIs, and $K \ll min(M, N)$ are the latent factors. Thus, the user-POI score is modeled as an inner product between the U and P matrices [Koren et al., 2009].

Hu et al. adapts the traditional MF algorithm for implicit feedback [Hu et al., 2008]. Weighted regularized matrix factorization (WRMF) is a popular model-based technique for one-class rating [Pan et al., 2008; Kang et al., 2016]. This method learns from implicit feedback for item prediction and implements a weighting matrix to differentiate rating activities and unobserved ones.

Rendle et al. proposed a Bayesian Personalized Ranking (BPR) for matrix factorization [Rendle et al., 2009]. BPRMF uses a stochastic approach to sample negative items for each user, and reduces the computation time significantly [Dasgupta, 2016]. As state-of-the-art collaborative filtering approach for implicit feedback, the BPRMF is widely employed as a baseline in several studies [Kang et al., 2016; Liu et al., 2017; Guo et al., 2017]. In contrast with other CF recommenders that are rating prediction oriented, BPRMF learns the ranking order through a pairwise classification. A variation of BPRMF is the weighted version WBPRMF. By weighting each POI pair with randomly-generated values, the WBPRMF attempts to promote low scored POIs.

Context-Aware POI Recommendation

As described in Section 2.2, besides the geographical constraints, POI recommendation task also consists of different kinds of information. Contextual information may detail more about the user's check-in behavior, for example, climate characteristic at the POI check-in or whether a user is with a friend.

As shown by Yu and Chen [2015], studies are being conducted to analyze how contextual information can improve a POI Recommendation. Gao et al. [2013] proposed a novel recommendation framework based on temporal context, achieving a good precision only considering when users have made check-ins (i.e., weekday/weekend, the day of the week). Likewise, Bannur and Alonso [2014] did an extensive study to analyze different temporal characteristics, and present how different seasons impact in POI visitations through the United States.

Ye et al. [2011] proposed a model to incorporate three factors to collaborative recommendation. USG models user preference (U), social influence from friends (S) and geographical influence from POIs (G) for POI recommendation. User preference adopts the traditional user-based collaborative filtering approach. Social influence is a friend-based collaborative filtering which based on similar friends tastes produce POI recommendations. Last, Geographical influence estimate the probability of the user visits a POI based on his or her historical check-ins.

Noulas et al. [2012] examine POI discovery behavior in two LBSNs, and shows that 60% to 80% of users' visits are in POIs that were not visited in 30 previous days. They proposed a model based on personalized random walk considering Collaborative Filtering (CF) approaches, social filtering and spatial filtering.

Hu et al. [2014] considers five features and its influences to enhance recommendations. In particular, this work analyzes the neighborhood influences in POI recommendation, which takes into account if the POI neighborhood and their popularity impacts in a better recommendation.

Recently, throughout an extensive evaluation over several POI recommenders, Liu et al. [2017] stated IRenMF, GeoMF and RankGeoMF as top-3 best models for POI recommendation. Those three recommenders are based on implicit feedbacks models and exploit geographical information to POI recommendations.

Instance-Region Neighborhood Matrix Factorization (IRenMF) is based on WMF. IRenMF exploit two levels of geographical neighborhood characteristics, (i) nearest neighboring locations tend to share more similar user preferences (locations-level influence); and (ii) POIs in the same geographical region may share similar user preferences (region-level influence) Liu et al. [2014].

Like IRenMF, GeoMF also is a geographical WMF model. Lian et al. [2014] incorporated the geographical context in GeoMF by modeling the influence propagation of the users' activity regions on geographical space. Moreover, because of the limitation of WMF, it is not easy to generalize the method to other types of context information.

Li et al. [2015] developed a ranking-based geographical factorization method to be very flexible to incorporate additional contexts. Rank-GeoFM makes a pairwise comparison of POIs and measures incompatibilities between the inferred ranking and the ranking produced by a factorization model. Additionally, Rank-GeoFM introduces an extra matrix factorization to incorporate the geographical context for POI recommendation. Despite those three approaches are equally good, Liu et al. [2017] described Rank-GeoFM as the one which usually performs the best POI recommendations.

Following Liu et al. [2017], we evaluate several POI recommendation techniques in this dissertation. We perform a breakdown analysis of eight state-of-the-art recommenders through different levels of sparsity and neighborhood densities. Also, we obtain important findings of the performance of these algorithms throughout our analysis. In addition, we propose a new task and evaluation methodology by employing the static pruning concept from the web search engines to the POI recommendation problem.

2.3 Static Index Pruning in Web Search Engines

The query processing is a critical task in Information Retrieval (IR) since it is an online step of search engines, where the user informs the query and expects to retrieve relevant documents for that query. The main objective in query processing is to estimate the relevance of a set of documents to a given query, rank these documents and present a small set of most relevant documents to a user. Cambazoglu and Baeza-Yates [2011] said that *"a slow query processing system resembles a library with a long line of impatient clients waiting at the front desk"*.

Therefore, in order to increase the query processing performance, Carmel et al. [2001] introduced the concept of *static index pruning*, removing potentially less relevant entries from the index. In this study is proposed a term-based pruning as a strategy to reduce the set of documents. Thus, by filtering the documents a priori helps to reduce the computational cost of the second phase of query processing. In the second phase, the documents selected in the first phase are ranked by a complex ranking model [Cambazoglu and Baeza-Yates, 2011]. The final ranking is obtained sorting the documents by their scores computed in the second phase.

Like static index pruning in search engines, in this dissertation, we propose a static pruning for POI recommendation. In addition, we propose the location-based strategy for static pruning, where we exploit the geographical influence of the POI recommendation task to prune out-ranged POIs from the recommendation. We expect that by pruning POIs outside the user's geographic range reduces the computation cost of predicting scores for all POIs in the target city without reducing the effectiveness of POI recommenders.

2.4 Summary

This chapter introduced recommender systems and their foundations. Section 2.1 presented the definition of the term "Recommender Systems" and discussed its functionality and history. In Section 2.2, we approached the POI recommendation problem, showing its relevance in LBSNs and for academic research. We discussed the new properties and challenges in POI recommendation, compared with traditional recommendation problems. Also, we presented the techniques employed by POI recommenders. In the vast majority, context-aware POI recommenders exploit the geographical influence to improve recommendations. The literature mentions a common human mobility behavior through the check-ins, whereby the user-POI distance impacts the likelihood a user will be interested in a POI [Bao et al., 2015]. In light of these findings, we observe that geography should be piece of the POI recommendation task instead of just a feature for POI recommenders. Thus, to guide this dissertation we raise two general questions:

1. What if, to exploit such a locality pattern, we consider the geographical context built into the POI recommendation task?

2. What if we restrict the recommenders to work with data in the user's vicinity?

In Section 2.3, we described how we can apply the static index pruning from search engines into POI recommendation. Moreover, this is the first work investigating the static pruning impact for POI recommendation. In this chapter, we only targeted a scientific perspective on POI recommender systems and identified a research gap. The next chapter describes the construction of the dataset that we use in this dissertation and its challenges.

Chapter 3

Nearby Point-of-Interest Collection

This chapter presents a new test collection for context-aware nearby POI recommendation as well as the methodology to collect the data. Section 3.1 describes the framework used for data collection and the challenges involved in creating the POI collection. In Section 3.3 we characterize the nearby POI collection and present the contextual information available in the collection.

3.1 Check-ins Collection

To build the POI collection, we collected the users' check-ins from Foursquare, nowadays one of the most popular LBSNs. In July 2017, Foursquare reported having a community with more than 50 million people using it with an average of 9 million check-ins a day, suppressing a total of over 12 billion check-ins[1]. However, once these check-ins are not publicly available by Foursquare, the LBSN allows the users share their check-ins via Twitter, where they are publicly available.

The Nearby POI Collection contains users' check-ins spread over one year from the entire world. The following section describes the framework and methodology to collect the users' check-ins from Twitter. Next, we introduce the Swarm (Foursquare's application for check-ins) and the filtering processes to guarantee congruence and consistency of check-ins in collection.

3.1.1 Twitter

The Twitter is an online news and social networking service where users post and read messages restricted to 140-characters, which are called *tweets*. This social network

[1] `https://foursquare.com/about`

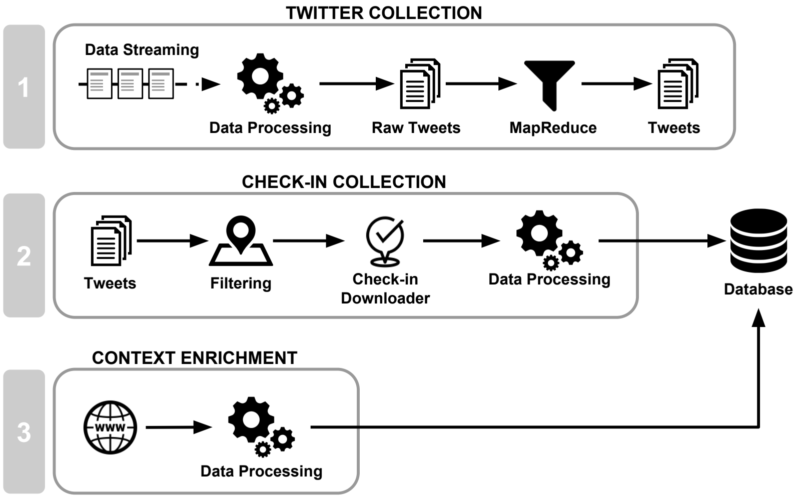


Figure 31: The collector's architecture.

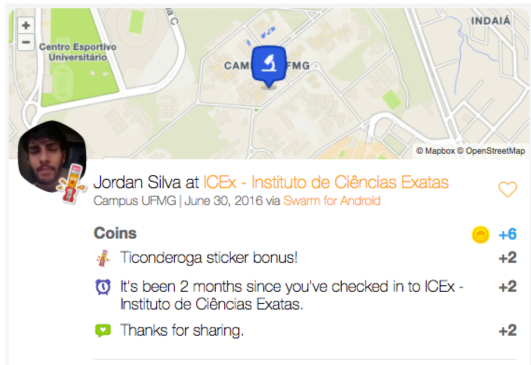
provides an API to access to the global stream of public tweets, where we can receive messages filtered by specific locations or a set of terms. Unlike Foursquare that does not allow to collect their check-ins dataset, on Twitter we were able to collect real-time public tweets. Despite the Foursquare not publicly provide the users' check-ins, the users are able to share their check-ins as they please, e.g., social networks (e.g., Twitter, Facebook), through e-mail, etc. Therefore, we can collect these check-ins that are shared from Foursquare to Twitter.

In order to build the POI collection, we collected real-time tweets related to Foursquare check-ins a whole year. We retrieved an average of 500,000 daily check-ins around the world using the Twitter Streaming API[2] and filtering by the following terms: *foursquare com, 4sq com, swarm com, swarmapp com*. These terms represent the Foursquare and Swarm websites addresses, respectively and they are present in every shared check-in through Twitter, linking the tweet to a check-in on Swarm website (Figure 32).

In the Figure 31, we can observe the first box (1) the whole workflow to collect the tweets divided into five steps. First, we receive the data streaming with tweets (step 1) and process them (step 2). Next, we store all those received tweets in raw format (step 3). Currently, we do not need the entire information provided by the raw tweets[3], thus, we remove unused information from tweets (step 4). Finally, we store the final version of the tweet (step 5) and we summarize the description of all fields in Appendix A1. Despite some tweets provide the *place* and *coordinates* information, they only represent a macro-geographic location of tweet (e.g., city, state, country).

[2]`https://dev.twitter.com/docs`
[3]`https://dev.twitter.com/overview/api/tweets`

(a) Twitter



(b) Swarm

Figure 32: The Foursquare shared check-in on Twitter.

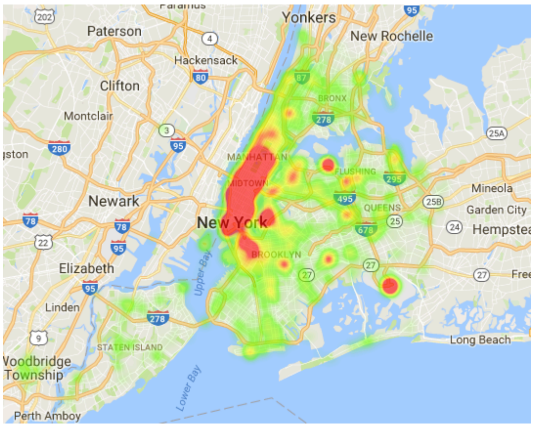
Therefore, to obtain the tweet checked POI we need to perform one more collection from *swarm url* field. In the following section, we describe the location-based social network Swarm and the process to collect the user's check-in.

3.1.2 Swarm

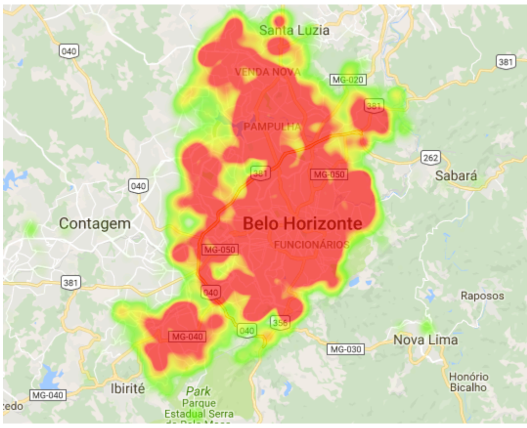
The Swarm is a mobile app that allows users to check-in to a given location, make plans with friends and see who is nearby. The application supports attach photos or stickers to check-ins and allows broadcasting of check-ins to other networks including Facebook and Twitter. Raised from the Foursquare, Swarm was a piece of the social network Foursquare at the beginning, which in May of 2014 became a stand-alone application. Currently, the location and check-in data collected in Swarm are used to improve user's recommendation in Foursquare. Like Foursquare, Swarm does not provide the users' check-ins data, however, if the check-in is shared and public, we can obtain the *url* to access the check-in information website (Figure 32b).

In the previous section, we performed a collection through Twitter obtaining the required *swarm url* to collect the checked POIs. Since the collected tweets (of check-ins) does not have the POI information but have a Swarm URL, was necessary retrieve these Swarm pages to get the content of check-in. However, we collected an average of 500,000 check-ins per day on Twitter, thus, performing all these URL request on Swarm's servers could take a long time. To reduce the request time for those check-ins, we perform a filtering of the tweets as presented in the second box (2nd box) of the check-in collection workflow in Figure 31. The check-in filtering (step 2) discards all non-geolocated tweets. By filtering these check-ins, (i) drastically reduces the amount of tweets and (ii) it allows to retrieve the check-ins by city, which reduces the amount of requests we need to make on Swarm.

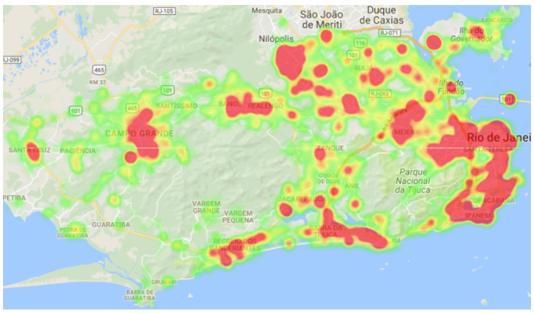
We choose three cities three large cities in the Americas to start our POI collection. We choose the New York City (NYC) due to its great cultural diversity as well as being a world-known city. Like NYC, Rio de Janeiro is a well-known city around the world, also is one of the most touristic cities in Brazil. Lastly, Belo Horizonte is the capital of the state of Minas Gerais in Brazil, the eighteenth largest city in the Americas, and also the city where this research is carried out. Thus, to download the user's check-ins (step 3), we select geolocated tweets in these cities and request their swarm urls. The swarm page has a JSON data (in source code) with the check-ins description, i.e., information about the user, the checked POI, badges earned, device used in check-in, timestamp, etc.



(a) New York



(b) Belo Horizonte



(c) Rio de Janeiro

Figure 33: Check-ins coverage at the target cities.

Once the Swarm check-ins are collected, we perform several procedures (step 4) to sanitize and verify the congruence and consistency of the data. Through this data processing, we can detect and correct invalid events that may occur during the data acquisition process. For instance, users can share their check-ins at any moment. Thus, we may find check-ins before the collection date. Also, the tweet may be geolocated in a city (New York City, USA) and the swarm check-in in another city or even a different country (Rio de Janeiro, Brazil). Thus, to remove these noisy check-ins, we consider only check-ins up to 60 km from the downtown area of the target city and that were checked in the period of the collection. As observed in Figure 33, we obtain a high coverage of the geographic perimeter of collected cities.

3.2 Context-Aware Enrichment

A person can estimate an uncountable amount of *variables* to decide whether she or he will go [Ferreira et al., 2015]. The combination of these *variables* create unique characteristics that define the likelihood of a person visiting a place. For example, one of those variables might indicate whether the place is suitable for kids (social); or provide information on whether the place is good at night (temporal). These *variables* are defined as contextual information about the user's preferences, the POI, environment, friendship, etc. As mentioned earlier, POI recommenders leverage contextual information to improve the POI recommendation. Thus, to provide such contextual information for POI recommenders, we performed an enrichment of the Nearby POI collection adding four different contexts: (1) temporal context; (2) climate information; (3) social influence; and (4) the user-location before the check-in. The following sections will give an overview of these contexts and the importance of them for POI recommendation.

Table 31: The contextual features available in Nearby POI collection

type	**feature**	
geography	user location	nearby POIs
temporal	daytime time is weekend	season weekday
climatic	temperature apparent temperature icon precipitation probability	summary cloud cover humidity

3.2.1 Temporal Context

The temporal influence is a fundamental feature for POI recommenders since it enables several discoveries of the users' patterns through the time of check-ins. The users' check-ins preferences tend to vary at different time slots as observed in existing studies related to POI recommendation [Cho et al., 2011; Yuan et al., 2013, 2014]. To illustrate, a user may prefer going to a restaurant at the evening and bars late at night, and then POI recommenders can prioritize recommendations for bars rather than restaurants at night.

Table 31 describes the temporal context available in the collection as five features as follows: check-in time, daytime (i.e., morning, evening, afternoon, night), season, weekday and if it is a weekend or not. By adding descriptive information about the check-in time in our collection, we facilitate to use the temporal context in the POI recommendation task.

3.2.2 Climatic Context

The climatic context has a strong influence on POI recommendation, for instance, if the climatic conditions are not appropriate for visiting outdoor attractions then the POI recommenders can exploit this information and suggest other places to visiting [Meehan et al., 2013]. Also, the climatic conditions can affect the distance that users are willing to travel to visit a place. However, few studies have been conducted using climate as a context for POI recommendation [Meehan et al., 2013; Trattner et al., 2016].

To encourage the exploration of this important context, we included the climate information for each user's check-in in our collection, facilitating the use of the climatic context in more studies. We collected the weather information such as temperature, humidity, and cloud coverage through Dark Sky API[4] (Table 31). The full list of climate features provided in the collection is detailed in Appendix A2.

3.2.3 Social Context

In the physical world, the discovery of a new place usually comes from word-of-mouth. In fact, social influence helps not only for a POI recommendation but identifying patterns and citizens' behaviors. An analysis of social groups and citizens' activities can contribute to the development of the society as a whole. For instance, citizens' behaviors help to anticipate an intense traffic in a particular neighborhood (e.g., Waze); citizens traveling a long distance for to school may mean the need for a closer school. Following this idea, a collection providing more social data is necessary, i.e., social relationships, citizens' profile, activities.

[4] `https://darksky.net/dev/`

Toward support future studies using social influence, our collection provides 4,273 social interactions, which represents a relationship between people who checked-in at the same place in a 1-hour interval at maximum. Also, 462,641 tips (New York City and Belo Horizonte) contributes as an electronic-of-mouth recommendation, that through tips and reviews people can express his or her feelings and suggestions [Moraes et al., 2013].

3.2.4 Geographical Context

The users' location plays as a central feature of the POI recommendation [Zheng, 2012]. Despite that, this feature is not available in the public test collections for POI recommendation. Due to limitations, the users' geographic information at the moment of the recommendation is not provided by the LBSNs. Since the users' location is missing from the POI collections, most of the POI recommenders overlook to this important feature.

Currently, to cope with the leak of the users' location, POI recommenders considers users' location as the users' home. The home location can be estimated using the users' check-ins history [Cheng et al., 2011]. This approach assumes that the user requests a recommendation always at home, as well as considers any POI as a candidate to be recommended.

Unlike the home location approach, we cope this problem assuming that most users' requests a POI recommendation nearby to him or her [Ye et al., 2011; Yuan et al., 2013]. Analyzing the CDF of the check-ins distance in our collection, Figure 34 shows that 60% of the check-ins in New York City are within 5 kilometers from the users' home[5], and more than 80% of the check-ins are within 10km from the user's location. Since most of the check-ins are near the user, our analysis is in agreement with previous studies stating that the user acts in geographically constrained areas and chooses to visit nearby POIs [Zhao et al., 2016].

In light of these findings, we build a collection that each check-in has a *given user location*, and the location is defined randomly at a distance d from the visited POI. Furthermore, since most POIs are out-ranged and are not plausible candidates to be recommended, we provide the Nearby POIs in the collection. It is a set of 10 or more candidate POIs, ordered by the POI distance from the user location named as users' neighborhood (Table 31).

[5]In this analysis, the users' home is defined by the centroid from the users' check-ins history

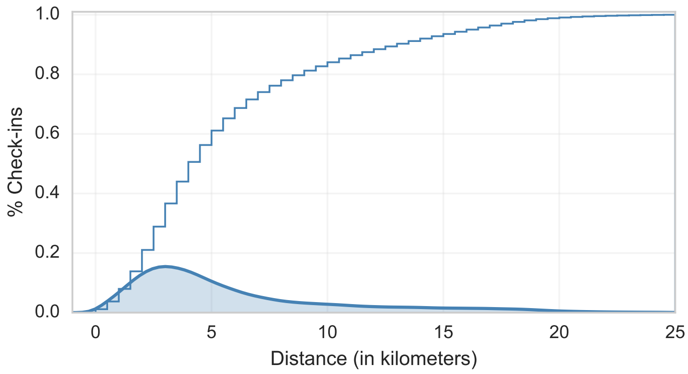


Figure 34: Distribution check-ins by distance in the New York City.

3.3 Dataset Characterization

The context-aware Nearby POI test collection is based on the check-ins collected from the Foursquare as mentioned in Section 3.1. The test collection covers check-ins from New York City (US), Belo Horizonte (BR) and Rio de Janeiro (BR) between Aug. 2015 to Jul. 2016. Table 32 summarizes the statistics of the Nearby POI collection.

Table 32: Nearby POI Collection

city	**# users**	**# POIs**	**# check-ins**	**# social links**
New York (US)	26,357	46,663	515,861	3856
Belo Horizonte (BR)	4,850	33,245	389,605	417
Rio de Janeiro (BR)	18,394	38,552	481,164	-

In the nearby POI collection each check-in basically consists of the following attributes: user, user's location, time of check-in, the set of candidate POIs near the user's location, and the target POI. Also, we provide additional features such as the POI categories (Table 33) in an auxiliary structure to complement the check-in data. The full list of available features is detailed in Appendix A3. In addition, as reference for comparison, we describe the most popular collections for POI recommendation in Appendix A4.

Table 33: List of POI categories

Arts & Entertainment	Outdoors & Recreation	Food	Travel
College & Education	Professional	Shops	Other
Event	Residence	Nightlife	

3.3.1 Considerations

Although a large amount of data produced by users in the LBSNs, we faced several limitations during of the collection process to obtain the user's check-ins from the Foursquare. First, the Twitter Streaming API (in the free version) limits to collect only 1% of total volume of produced tweets. Thus, this limitation restricts us from collecting all Foursquare check-ins shared through Twitter. Despite that, we retrieved an average of 6% of Foursquare check-ins ($\approx 500k$ check-ins per day) comparing with the 9 million of daily check-ins reported by Foursquare. Second, LBSNs mainly consist of young people who frequently use social networking services [Yang et al., 2015]. Hence, citizens' behavior may be biased because of the users of LBSNs. Finally, we also faced a limitation of the representativeness of check-ins due to fewer users share their check-ins on Twitter [Long et al., 2013].

Another consideration is that, despite we have presented three cities in this collection, we will use in this dissertation only the New York City collection. As aforementioned in Section 3.1.2, we choose the New York City due to its great cultural diversity as well as being a world-known city. Likewise, we consider only the temporal and geographical context in the analysis of this work. We provide the other two cities as well as social and climatic contextual information as an extra of this collection. Nonetheless, we intend to use those additional data in future research.

3.4 Summary

In this chapter, we described our framework to collect and build the context-aware Nearby POI collection as well as the difficulties that we faced creating the collection. In Section 3.3, we characterized the collection and described four additional contexts available in the collection. Also, we highlighted the limitations of our collection. This chapter addressed the first contribution of this dissertation, a geography-aware test collection which includes several geography-constrained test cases, enriched with temporal, social and weather contexts.

Chapter 4

Nearby POI Recommendation

POI recommendation is an important service to LBSNs, helping in the suggestion of new places that the users are likely to be interested in visiting. The vast majority of studies about POI recommendation consider three main sources of information to build and evaluate a POI recommender: (i) the user; (ii) the venues visited by the user; and (iii) the set of candidate venues for recommendation. Using this information, the recommender suggests a new POI $p \in \mathcal{P}$ to a user $u \in \mathcal{U}$ by learning the users' preferences from the POIs that they have visited ($P_u \subseteq \mathcal{P}$).

In order to evaluate the accuracy of the recommendations, these studies employ an evaluation framework which (i) implements a traditional cross-validation not ensuring a chronological order the check-ins; (ii) allows to target multiple relevant places during the evaluation phase; and (iii) requires the scoring of all POIs in the dataset. Arguably, this *traditional evaluation methodology* does not model realistic POI recommender scenarios as it overlooks important features such as users' location and geographical context of a venue [Yuan et al., 2013]. The user's location plays an important role in LBSNs since the recommenders typically start suggesting POIs for the user's visitation from his or her location. Disregarding the user's location at the time of the recommendation, the traditional evaluation framework uses the entire collection of POIs $\mathcal{P}$ as candidates, which may result in noisy recommendations. Also, the score prediction for the whole set of POIs results in an unnecessarily high computational cost, since most POIs would be outside the geographic range the user would be willing to visit, thus, not being plausible candidates. To cope with these limitations, we propose the Nearby POI recommendation task, where the geographical context is used to restrict the recommendation to nearby POIs.

In contrast with the traditional evaluation frameworks that consider any POI as a candidate, the *Nearby POI recommendation* task benefits from the users' location to prune out-ranged POIs, removing noisy POIs of the recommendation. The Nearby POI recommendation employs Tobler's First Law of Geography which states that *"Everything is related to everything else, but near things are more related than distant things"* [Tobler, 1970]. Tobler's First Law of Geography implies that users prefer to visit nearby locations rather than distant ones [Yu and Chen, 2015]. Thus, considering the importance of the geographical context in the POI recommendation, the Nearby POI recommendation task explores the *user's neighborhood*, i.e., POIs surrounding the user's location.

The following sections discuss two core properties of the Nearby POI recommendation task. Section 4.1 discusses the *user's location* definition and its uses for the task. Section 4.2 describes the *neighborhood* and the potential benefits of using a nearby set of candidate POIs. Last, Section 4.3 shows a contrast between the characteristics of the Nearby and Traditional POI recommendation tasks.

4.1 User's Location

The user's location is an important feature of the POI recommendation in LBSNs. Likewise, the users' location is at the core of the Nearby POI recommendation task, since it is essential to establish the set of candidate POIs for recommendation. As introduced in the Section 3.2.4, LBSNs only provide the check-ins location. Consequently, the available test collections for POI recommendation do not provide the user's location at the moment of recommendation. Despite this, previous studies addressed this issue with three different strategies: (i) ignoring the user's location; (ii) assuming the user is at home (the location of which must be inferred); and (iii) assuming the users' location is the last POI that they visited, which is particularly exploited for recommending sequences of POIs.

The strategy of *overlooking the user's location* cannot be used by the Nearby POI recommendation since it requires the user's location to build the user's neighborhood. If we just ignore the user's locations, we retain the same problem of the traditional POI recommendation task, which considers every POI in the dataset as a candidate. The next strategy - *inferring user's home* - assumes that users always get a POI recommendation from their homes. In the same way, the POI recommenders need to consider the entire POI collection as candidates since the users at some point will visit places far from their homes. Another concern about inferring the user's home from his or her historical check-ins is that his/her home could be mislocalized. Arguably, the *sequential POI recommendation* strategy is the one that makes the most sense among the others, using the last known user-location for a recommendation. However, we discard this strategy because it implies that the interval between the user's check-ins needs to be in a small time window. Thus, this strategy does not fit entirely for a real-world scenario, where there are a large number of users with few check-ins and users who have very sparse check-ins.

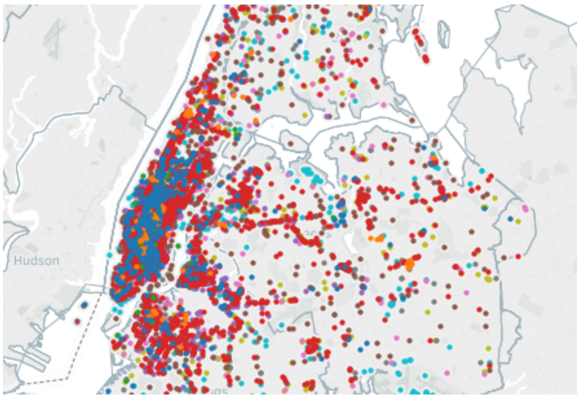
Section 3.2.4 introduced a POI collection containing user's location for each check-in that the users did. To simulate the users location in our proposed evaluation methodology, we randomly place the user in the vicinity of the venue where he or she actually checked-in. To estimate the user's location we define the maximum distance r that the user can be from the visited POI, then choose a random position within that distance r. Once the user is near the visited POI, this eliminates the need of considering the entire POI collection as plausible candidates. The advantage of this strategy is that we can prune all out-ranged POIs, thus, not being plausible candidates [Ye et al., 2011; Yuan et al., 2013; Yu and Chen, 2015]. Also, employing this strategy, we no longer need to infer the user's home through his or her check-ins or use the sequential check-ins strategy. Moreover, this strategy allows parameterizing our evaluation by using different maximum distances r. In the extreme, with $r \to \infty$, the Nearby POI recommendation task is equivalent to the Traditional POI recommendation task. While we experiment with fixed r values in our evaluation in Section 6, customized values could be used to reflect users with different willingness to travel long distances, as well as neighborhoods with different densities. We leave this investigation for the future.

4.2 User's Neighborhood

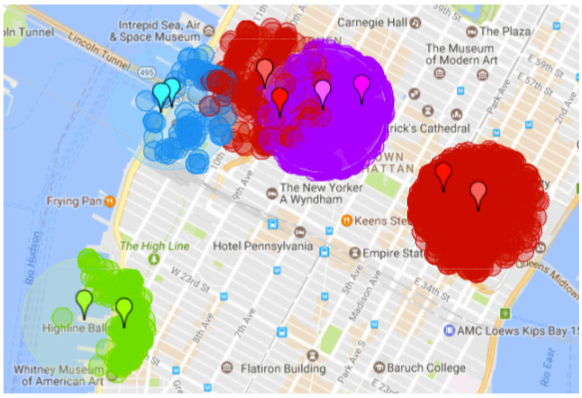
User's neighborhood plays a prominent role Nearby POI recommendation task, as it allows pruning the number of candidate POIs for the recommendation. A *user's neighborhood* is an area that the user is willing to explore. For instance, an area might be a street in a city, a borough, neighborhood, or even the whole town. Still, the larger the area to explore, the greater the number of POIs within it. However, in order to reduce the number of POIs, we apply Tobler's First Law of Geography which suggests that *the probability for a user to visit a venue is closely related to the distance the user needs to travel to reach the venue* [Liao et al., 2016]. Thus, we assume in the Nearby POI recommendation task, that people will visit POIs in a restricted area near to his or her location, determined as discussed in Section 4.1.

Therefore, a *"neighborhood"* is the vicinity of the user restricted by a radius d from his or her location. Thus, the set of the candidate POIs will include any POI

Figure 41: The number of candidate POIs per recommendation task. (a) The traditional task considers every place in the city as a possible candidate. Each point denotes a POI. The colors express different POI categories. (b) The Nearby task explores only the users' neighborhood, from the users' location. The major circles represent the size of the neighborhood and the minor circles are the candidate POIs; the marker on the circle's center is the user's location and other the relevant POI. The colors mean different check-ins.



(a) Traditional POI recommendation task



(b) Nearby POI recommendation task

located within the neighborhood range. Note that for the POI recommender to be able to recommend the relevant POI, this POI must be within the neighborhood. Thus, $d \geq r$, where r is the distance from the user to the relevant POI (Section 4.1). To illustrate, Figure 41 shows a contrast between the set of candidate POIs of the Traditional and Nearby POI recommendation tasks. Figure 41a presents a unique user check-in and the set of candidate POIs which are the colored circles on the map. Figure 41b shows five different check-ins and their candidates POIs by color. Each check-in has a major circle that is the user's neighborhood; a light marker in the center of the neighborhood represents the user's location; the other marker with the same color of the neighborhood represents the relevant POI; the minor circles within the neighborhood are the candidate POIs. Besides the difference of the number of candidate POIs in the two tasks, we can observe in Figure 41b that although the neighborhoods are the same radius d, the number of POIs within each neighborhood may vary due to its density, which arguably adds to the realism of the task by producing test cases of varying difficulty. In Section 6.2.2, we will analyze how different POI recommenders perform across test cases of different densities.

4.3 Overview

To highlight the main differences of the Nearby POI Recommendation task from the traditional POI recommendation task, we characterize the tasks in the following four components: (i) the travel distance; (ii) the user's location; (iii) the set of candidate POIs; (iv) the context-awareness.

The travel distance component defines how far a place can be from the user for a recommendation. While traditional POI recommendation evaluation do not limit the distance of the recommendation, the Nearby POI recommendation task has a distance limitation a priori which can be parameterized. Thus, traditional POI recommenders explore the whole city, and the Nearby POI recommendation explores only a restricted area for a recommendation. Note that in the Nearby POI recommendation we can set the distance limitation to reach the entire city as well.

User's location, typically determined via global positioning system (GPS) coordinates or alternative geolocation approaches, is at the core of LBSNs as a reference point for POI recommendations [Zheng, 2012; Yuan et al., 2013]. To address the missing user's location in check-in datasets, traditional approaches infer users' homes from their previous check-ins or even do not exploit the user's location. Nearby POI recommendation task introduces a new strategy to cope with the user's location problem, randomly choosing a user's location near - i.e., at a parameterized distance - the visited POI. As the Nearby task considers that the user will visit nearby POIs, the POI recommenders can learn the user's distance preference and configure this range per user.

Another essential component of the POI recommendation task is the set of POIs that will be recommended. As the traditional POI recommendation approaches allow users to explore the whole city, there will be no constraints on the set of POIs. Hence, all places in the city will be considered potential candidates for the recommendation. On the other hand, the Nearby POI recommendation task has a distance constraint that limits the potential candidates to those in the user's neighborhood. Reducing the set of candidate POIs has the potential to alleviate the high computational cost of predicting scores for a large number of POIs, as well as training effective POI recommendation models.

Last, context-awareness has been shown to produce better POI recommendations [Trattner et al., 2016; Gao et al., 2013]. For instance, Li et al. [2015] present that leveraging information about the characteristics of the geographical surroundings of a venue improves recommendation. Because the geographic context is inherently important to POI recommendation, we argue that it is more realistically handled as a constraint rather than as a ranking criterion. Indeed, unlike the Traditional POI recommendation task, Nearby POI recommendation enforces a geographic constraint by construction. As a result, we can better assess the effectiveness of alternative recommenders and even alternative contexts (e.g., time, climate, companion) regardless of geography.

4.4 Summary

In this chapter, we introduced the Nearby POI recommendation task. We discussed the differences between the Nearby POI and Traditional POI recommendation tasks. Also, we proposed a new strategy to cope with the problem of missing user's location data for evaluating POI recommenders and compared it with traditional approaches from the literature. Later, we presented the concept of user neighborhood as an important characteristic of our proposed task, as it allows reducing the recommendable POIs to the user's neighborhood. Finally, we contrast the key features of both tasks.

Chapter 5

Experimental Setup

In this chapter, we describe the evaluation setup, baselines, and additional information of the dataset used for the analyses in Chapter 6. In particular, our experiments seek to model a more realistic scenario for POI recommenders paying attention to user's location and the geographical context of the venues. As introduced in Chapter 1, we aim to answer the following research questions of this dissertation:

- **RQ1:** How does location-based pruning impact POI recommendation effectiveness and efficiency?
- **RQ2:** How do different levels of data sparsity affect the effectiveness of POI recommendation methods?

In the remainder of this chapter, Section 5.1 formalizes the POI recommendation problem. Section 5.2 details the test collections, whereas Section 5.3 shows the evaluation framework used in our experiments. Lastly, Section 5.4 overviews the POI recommenders used for our experimental analyses.

5.1 Problem Definition

As discussed in Chapter 4, we model the POI recommendation task to a ranking problem which has a set of candidates to be ranked. In particular, each test case is denoted by a tuple $<u, l_u, t, \mathbf{C}_{l_u}, v>$, where u is a target user, l_u is the user's location (randomly defined) at time t, C_{l_u} is a set of candidate POIs in the vicinity of u (i.e., within a distance d from l_u), and $v \in C_{l_u}$ is a relevant POI (i.e., the POI where the user last checked in). In contrast with the traditional POI recommendation that may allow many relevant items for a single test case, we have only a single relevant venue for each test case. For this reason, our task is interpreted as a ranking problem since the objective is to rank the relevant POI v ahead of all other POIs in $\mathbf{C}_{l_u}$. Table 51 summarizes the previous notations.

Table 51: List of notations

$\mathcal{U}$	the set of users $u_1, u_2, \cdots, u_m$
$\mathcal{V}$	the set of POIs $v_1, v_2, \cdots, v_n$
t	the date timestamp that u checked in at v
r	the maximum distance that u can be from v
l_u	the user's location in the range r from v
d	the maximum range distance of the user's vicinity $d \geq r$
$\mathbf{C}_{l_u}$	the set of POIs $\mathcal{C}_{l_u} \subseteq \mathcal{V}$ at a maximum distance d from l_u

5.2 Dataset

We conduct the analyses on the Nearby POI collection described in Chapter 3. To evaluate the performance of the algorithms we use a sample of the New York City check-ins as our test collection. First, we exclude users with fewer than 10 check-ins from the test collection. Next, we also remove from the test collection every POI that received less than 10 check-ins. To perform a consistent evaluation, we sorted the check-ins chronologically and standardized the number of check-ins to 10,000 check-ins per month. The test collection statistics are summarized in Table 52

Table 52: Datasets

	# user	# POI	# check-ins	# training	# validation	# test	sparsity
New York	7,703	8,812	110,000	30,000[1]	5,000[1]	5,000[1]	0.9997
São Paulo†	6,226	16,530	88,000	24,000[1]	4,000[1]	4,000[1]	0.9994
Singapore‡	2,321	5,596	120,884	102,221[2]	-	18,671[2]	0.9906

[1]This statistic is measured by time window. 3 months for training, 1 month for validation and 1 month for test.

[2]The training and test statistics was not divided into time windows.

† Yang et al. [2015, 2016]

‡ Yuan et al. [2013, 2014]

We also evaluate and present the general results for two additional public collections - i.e., São Paulo and Singapore cities - in order to verify and discard any possibility of bias from our collection. However, a breakdown analyses and further experiments will be conducted exclusively in the New York collection. For the New York City collection we perform three tests, (i) using the user's neighborhood range constraint as 500 meters, (ii) user's neighborhood with a maximum of 1 kilometer and (iii) without restriction of the user's neighborhood range.

5.3 Evaluation Methodology

Our evaluation methodology aims to represent and simulate a realistic POI recommendation scenario. To this end, we perform an evaluation distributing the collection temporally. Our test collection divides into 8 time-windows for entire a year. Each time-window contains a period of 5 months of check-ins.

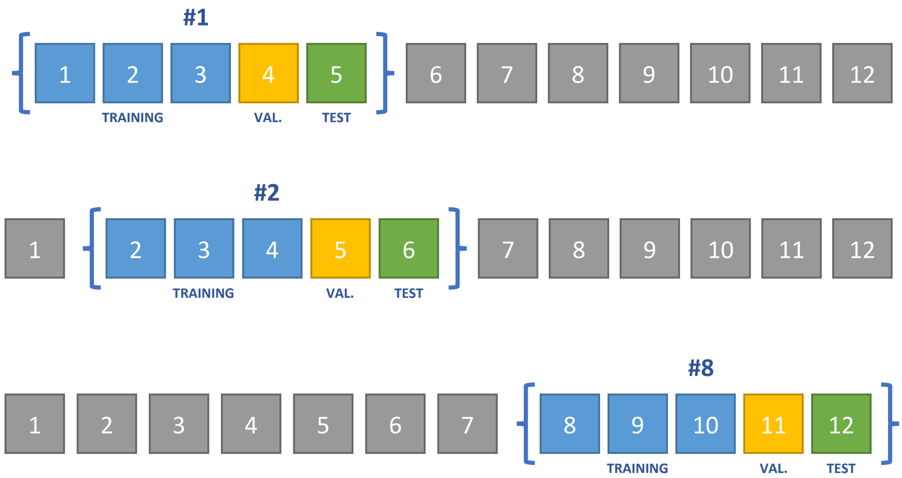


Figure 51: Test collection divided into 8-time windows. 5-boxes compose a time-window, and each box is a one-month check-in. The first three months inside the brackets, in the color blue, are the data for training. The next two boxes are validation (yellow) and test data (green), respectively. Above the time-window has a #<number> that represents the time-window tag.

Figure 51 shows the time windows inside brackets, consisting of 5 months of check-ins. Each time window has the three first months (blue) of data for training, the next month (yellow) for validation and the last one (green) for the test. The validation and test data are further filtered to retain only target users' check-ins which have never been visited before, i.e., a new POI for the user, reducing the total number of validation and test cases per month from the original number of 10,000 (the resulting numbers are described in Table 52). Thus, each evaluated fold may contain new users, new places and new recommendations, providing for realistic test cases encompassing non-cold-start as well as user, item, and system (i.e., user+item) cold-start situations.

Evaluation Metrics

To analyze the accuracy of the methods in the Nearby POI Recommendation task, we use the Mean Reciprocal Rank (MRR) metric. The Reciprocal Rank (RR) calculates the reciprocal of the rank position at which the target relevant POI was ranked. In turn, MRR measures the average of the RR results across all test cases, such that:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{5.1}$$

where $rank_i$ is the position of the relevant POI in the ranking, and Q denotes the set of test cases.

5.4 POI Recommenders

To evaluate the experiments of our proposed task and answer the research questions presented in this chapter, we analyze the behavior of eight methods for POI recommendation. Next sections describe these methods divided into non-personalized and personalized recommender systems.

5.4.1 Non-Personalized Recommender Systems

Non-personalized recommender systems use general information to compute the recommendations for the users. In non-personalized recommendation, all users receive the same recommendations, regardless of their preferences.

The popularity approach is based on recommending the top-N most popular items to users. Popularity-based recommendations are widespread as they are easy to compute and is offer a reasonable recommendation alternative to new users [Ye et al., 2013]. Another reason to employ such an approach is that users often wish to know about the most popular items [Burke, 2002]. In this work, we use the **Most Popular (MP) approach** based on the number of check-ins a POI has. Thus, the most visited places are prioritized in the recommendations. Note that most popular POIs may vary according to the neighborhood (i.e., the set of candidate POIs).

Geographic approach is also employed in this work as a non-personalized recommender system. Like the MP method, the POI suggestions are ordered by a single attribute of the place which is not personalized by user. The geographic approach provides a recommendation sorting the POIs by the distance, ranking at the top the places near the user. In addition, this approach serves as input for all recommenders in this work. Hence, we expect the Geographic approach to have the worst performance among the tested algorithms.

5.4.2 Personalized Recommender Systems

Personalized recommendation considers user's previous history of interactions with the available items when generating predictions [Khatwani and Chandak, 2016]. To conduct our experiments, we evaluate six well-known collaborative filtering recommenders from the literature and separate these recommenders into memory-based and model-based techniques. The memory-based approach uses the data (e.g., ratings, clicks, likes) to establish correlations between either users or items, in order to suggest an item that a user has never seen before. This technique relies heavily on simple similarity measures to match similar users or similar items. On the other hand, the model-based approach uses machine learning algorithms to predict how much a user will like an item. Through this approach, the algorithms can build models from the training collection that can predict the user's rating for a new item added to the system.

UserKNN and **ItemKNN** are neighborhood-based collaborative filtering (CF) techniques which are typical memory-based approaches for recommender systems. The neighborhood CF finds the top-k similar neighbors ranked by a correlation metric (e.g., Cosine, Pearson, Spearman, etc) and recommend these items. UserKNN is a user-based CF which looks for similar users (neighbors) and suggests what they like. Likewise, ItemKNN computes similarity to suggest items similar to those that the user likes [Sarwar et al., 2001].

Weighted regularized matrix factorization **(WRMF)** is a popular model-based technique for one-class rating [Pan et al., 2008; Hu et al., 2008]. This method learns from implicit feedback for item prediction and implements a weighting matrix to differentiate rating activities and unobserved ones [Kang et al., 2016].

BPRMF is the most popular ranking-based matrix factorization with Bayesian personalized ranking (BPR). BPRMF is a state-of-the-art collaborative filtering approach for implicit feedback [Rendle et al., 2009] and is also widely employed as a baseline in several studies from the literature [Kang et al., 2016; Liu et al., 2017]. In contrast with other CF recommenders that are rating prediction oriented, BPRMF learns the ranking order through a pairwise classification. Another method employed in this study is **WBPRMF** which extends BPRMF and weights each POI pair with randomly-generated values.

Rank-GeoFM is a state-of-the-art ranking-based matrix factorization method for POI Recommendation [Liu et al., 2017]. Like BPRMF, Rank-GeoFM makes a pairwise comparison of POIs and measures incompatibilities between the inferred ranking and the ranking produced by a factorization model [Li et al., 2015]. Furthermore, Rank-GeoFM is a context-aware POI recommender which incorporates the influence of the geographical context for POI recommendation. In particular, such algorithm learns the traditional users' preference rankings for POIs and models the geographical influence of neighboring POIs. To compute the recommendation score, Rank-GeoFM uses two latent matrices, the user preference matrix $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ to represent users' geographical preferences. Thus, the recommendation score is computed as follows:

$$y_{ul} = \mathbf{u}_u^{(1)} \cdot \mathbf{l}_\ell^{\top(1)} + \mathbf{u}_u^{(2)} \cdot \sum_{\ell^* \in \mathcal{N}_k(\ell)} w_{\ell\ell^*} \mathbf{l}_{\ell^*}^{\top(1)} \tag{5.2}$$

where the first term denotes the user-preference score, while the second term models the geographical influence score that a user likes a POI because of its neighbors. $\mathcal{N}_k(\ell)$ is the k-nearest neighbors of the POI ℓ and $w_{\ell\ell^*}$ refers to a distance-based weight between the POIs ℓ and ℓ^*.

5.4.3 Parameter Settings

We tune the parameters for each model via grid-search using our evaluation methodology discussed in Section 5.3. To find the optimal values that maximize the MRR in training data we use the first 5 months of the test collection (i.e., time-window #1). All experiments were performed in a PowerEdge R710 Intel(R) Xeon(R) CPU X5680 @ 3.33GHz 24-CORES with 64GB RAM using Ubuntu Server 12.04 LTS 64 bits.

5.5 Summary

In this chapter, we described the experimental setup that supports our investigations in Chapter 6. In Section 5.1, we recalled the definition of the Nearby POI recommendation task. Section 5.2 provided details of the test collection used in the analyses described in Chapter 6. In Section 5.3, we gave details of the evaluation framework which is intended to simulate a realistic POI recommendation scenario used by LBSNs. Last, Section 5.4 gave an overview of the POI recommenders employed for the experiments in Chapter 6.

Chapter 6

Experimental Evaluation

In this chapter, we evaluate the performance of several state-of-the-art methods for the Nearby POI recommendation task we proposed in Chapter 5. Section 6.1 presents the effects of location-based pruning on POI recommendation on test cases. In Section 6.2, we perform a breakdown analyses of the results, comparing the effectiveness of the algorithms for different sparsity conditions. Also, we further analyze the impact of various neighborhood densities in the Nearby POI recommendation task. Finally, Section 6.3 describes the Nearby Rank-GeoFM customization and contrast the effectiveness-efficiency impact of pruning vs. not pruning the candidate POIs on training data.

6.1 Pruned Test

The Nearby POI Recommendation task introduced the concept of user's neighborhood in Chapter 4. The user's neighborhood is an area near the user's location which the user is willing to explore. In short, the Nearby POI recommendation task restricts the set of candidate POIs to the user's vicinity rather than considering POIs in the whole city as possible candidates. Thus, akin to static pruning in search engines [Carmel et al., 2001], the nearby exploration uses the user's location to prune the number of candidate POIs for the recommendation. In particular, this section aims to answer a specific question related to $RQ1$: **"What is the global impact of pruning test cases?"**. To address this question, we analyze the effectiveness and the efficiency of the recommenders when they are restricted to make predictions only for POIs in the user's vicinity. In particular, we investigate the following hypotheses:

$H1$. The relative effectiveness of the recommenders changes due to the addition of geographic context to the task, with geography-agnostic recommenders closing the gap to geography-aware recommenders.

*H*2. The efficiency of the recommenders improves since pruning the POIs reduces unnecessary computation scores.

To analyze the consequences of pruning the set of candidate POIs in the test cases, we compare the accuracy results of a fixed set of POI recommenders under the traditional recommendation task as well as the Nearby POI recommendation task. We also perform a further analysis of the impact of employing different target radiuses for users' vicinity. Table 61 presents the accuracy of the recommenders on the test cases. The second column (#) of the table shows the position of each recommender relative to the others according to the MRR metric on the traditional task. The third column (traditional) shows the MRR accuracy of the recommenders considering the entire city as the users' vicinity to recommendation. The next columns represent the accuracy of the recommenders by restricting candidate POIs to the users neighborhood, using a range of 1 kilometer and 500 meters respectively.

Table 61: A performance comparison pruning the test cases with different neighborhood ranges in the New York City collection. # is the relative position of the method by the MRR metric. Neighborhood and user's location divides into 500 meters, 1 kilometer and traditional which does not use restriction of distance.

		Mean Reciprocal Rank		
method	**#**	**traditional**	**1 km**	**500 m**
MP	1	0.0845	0.1202	0.2365
BPRMF	2	0.0825	0.1126	0.2228
WRMF	3	0.0254	0.0604▼1	0.1341
Rank-GeoFM	4	0.0182	0.0440▼2	0.1066
UserCF	5	0.0169	0.0559	0.1259
WBPRMF	6	0.0157	0.0652▲3	0.1552
ItemCF	7	0.0025	0.0323	0.0889
Geographic	8	-	0.0331	0.0806

From Table 61, the Geographic method has the worst accuracy in general. As the base ranking for the other algorithms, we expected that Geographic recommendations would perform worse than the other algorithms. Also, the missing MRR result for traditional evaluation is due to the lack of users location information in this task, which is required to produce a distance-based ranking. Despite producing random recommendations (i.e., considering a random user location to ranking near candidate POIs first), Geographic method serves as a sanity check for the other approaches.

Item CF performs poorly when compared to other collaborative filtering methods, such as User CF. Employing solely the number of venues check-ins in the item similarity computation does not guarantee a good performance of the method. In contrast, using the users' feedback fits better for the POI recommendation. UserCF secured the fifth position in all results. Such results suggest that people have a predictable taste for places, visiting places that people similar to them have visited. In addition, the effectiveness of the neighborhood-based CF does not seem to be affected by different range sizes of user's vicinity.

On the other hand, WBPRMF had a significant improvement in pruned scenarios. Pruning the set of candidates by the user's vicinity promotes WBPRMF up to the third position in the rank.

Despite Rank-GeoFM being considered the state-of-the-art recommender for POI recommendation, it achieves only the fourth position in the traditional task. Worse still, by adding the geographical context to the task affects the Rank-GeoFM accuracy, dropping to sixth position. We speculate that the geographic authority of the Rank-GeoFM is reduced by considering only nearby POIs as candidates.

The WRMF demonstrated a better result than the other CFs recommenders such as WBPRMF and the KNNs. Although, WRMF also seems to be jeopardized in Nearby POI recommendation task, losing the third position to WBPRMF. This slight accuracy improvement of the WBPRMF over WRMF suggest that a probabilistic Bayesian Personalized Ranking (BPR) loss could be better suited for coping with POI recommendations. Despite the good results, both WBPRMF and WRMF showed a much lower performance compared to BPRMF, suggesting that a weighting of the check-ins frequency, in general, is not well suited for POI recommendation.

The BPRMF accuracy outperforms all the evaluated recommenders but Most Popular (MP). The non-personalized MP outperforms all evaluated recommenders regardless of the task. Therefore, we observe that the venue's popularity is the most significant feature that indicates whether a place is a good recommendation. In summary, suggesting that people tend to visit nearby popular venues. Furthermore, we believe that the poor accuracy of personalized approaches is due to the high sparse interactions as personalized approaches suffer from learning the users' preferences and underperforming on these conditions.

Results in Table 61 show that the relative effectiveness of the recommenders changes slightly as we constrain the set of candidate POIs to those in the users vicinity. Assuming geography by construction seems to improve context-agnostic recommenders such as WBPRMF, whereas the context-aware Rank-GeoFM recommender dropped some positions in the ranking order. Regarding the prediction efficiency of these algorithms, the Nearby POI recommendation task reduces the computational cost by pruning the candidate POIs and scoring k rather than all n available POIs, with $k \ll n$. In light of these findings, the results agree with our expectation, thus supporting $H1$ and $H2$. In addition, in Section 6.2 we conduct a breakdown analysis to understand the impact of different sparsity levels on the recommenders.

6.2 Breakdown Analyses

To further our understanding of the results presented in Section 6.1, we conduct a breakdown analysis using the New York City collection. In Section 6.2.1 we perform a breakdown analysis of the results, comparing the performance of the algorithms in different sparsity conditions. Section 6.2.2 analyzes how the contextual information (e.g., neighborhood areas and geographic data) impacts the Nearby POI recommendation task.

6.2.1 Check-in Sparsity

The scarcity of data is particularly severe in POI recommendation. Indeed, most users check in to very few places. In particular, our analyses aim to answer $RQ2$ by analyzing what test cases are impacted the most from the following sparsity perspectives: (i) item sparsity; (ii) user sparsity; (iii) user and item sparsity (i.e., system cold-start); and (iv) non-cold-start cases. Table 62 shows the accuracy of the algorithms split by the four result categories, as well as the total results.

Table 62: A performance comparison of different algorithms for the Nearby POI Recommendation task in the New York City collection using 500 meters as neighborhood range. Results are divided in five categories, i.e., three cold-start scenarios, a non cold-start, and the overall result (see Appendix C2 and C3 for the other collections).

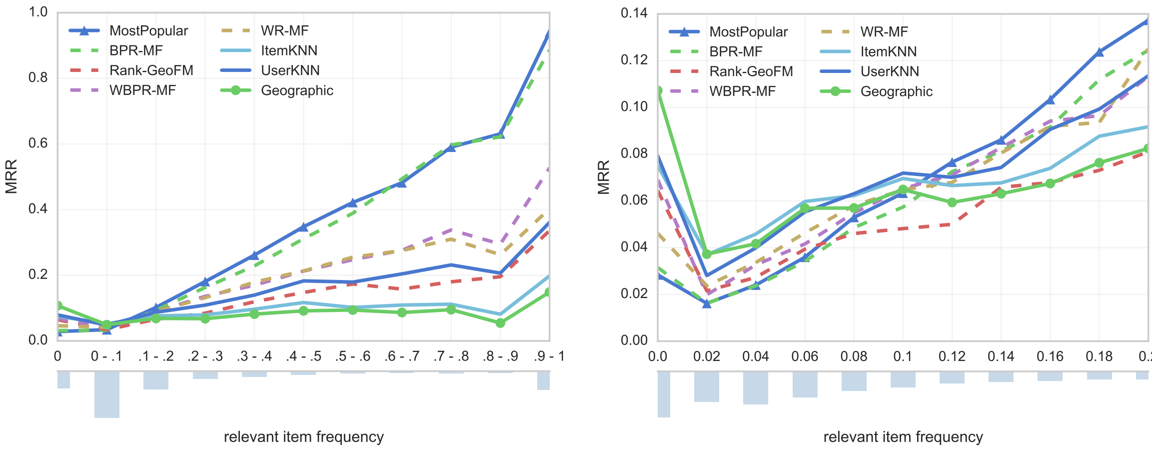
	Mean Reciprocal Rank				
	(100%) **total**	(10,6%) **item cs**	(28,5%) **user cs**	(3%) **cs**	(57,8%) **non-cs**
MP	**0.2365**	0.0292	**0.3626**	0.0255	**0.2235**
BPRMF	0.2228	0.0329	0.3542	0.0266	0.2033
Rank-GeoFM	0.1066	0.0641	0.1197	0.0652	0.1101
WBPRMF	0.1552	0.0665	0.1876	0.0777	0.1596
WRMF	0.1341	0.0306	0.0776	0.0993	0.1828
UserCF	0.1259	0.0733	0.0776	0.0993	0.1607
ItemCF	0.0889	0.0680	0.0776	0.0993	0.0978
Geographic	0.0806	**0.1095**	0.0776	**0.0993**	0.0758

Section 6.2.1.1 presents the difficulties that POI recommenders face when recommending with lack of feedback data. Section 6.2.1.2 studies user sparsity, analyzing the behavior of the algorithms through different users' check-ins history size. Section 6.2.1.3 discusses the problem in a system cold-start scenario, i.e., when we have no historical data about the user or the venue. Section 6.2.1.4 examines how the algorithms perform in non-cold-start scenarios, i.e., when previous information of the user and the venue is known.

6.2.1.1 Item Sparsity

POI recommenders are strongly affected by data sparsity. In this section, we analyze how the methods behave in different levels of *item sparsity*. The hypothesis we want to verify is that:

$H3$. Collaborative recommenders are severely harmed by item sparsity, more so for non-personalized recommenders.



(a) frequency from 0.0 to 1.0 (b) frequency from 0.0 to 0.2

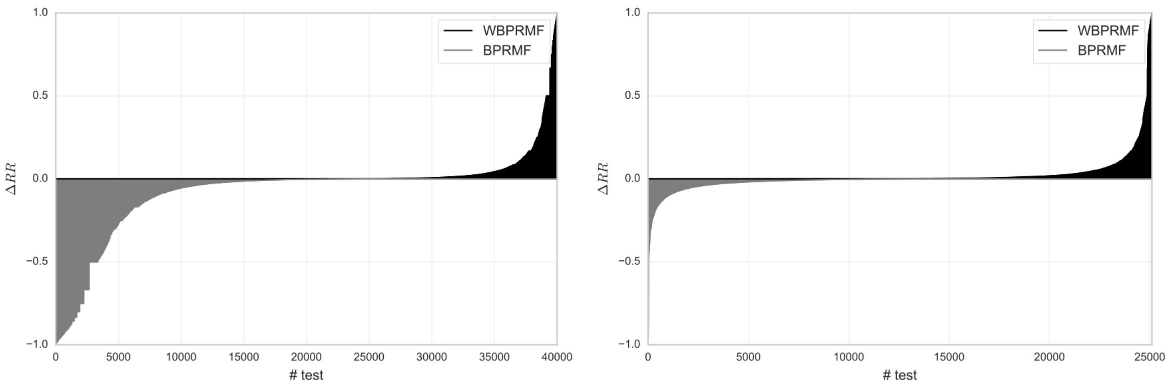
Figure 61: Relevant item frequency normalized by a set of candidate POIs. Higher values represent that relevant venue is most popular among the neighborhood.

In a nutshell, both non-personalized (e.g. Most Popular) as well as personalized (memory and model-based) collaborative filtering recommenders are significantly jeopardized in the absence of feedback data. Since the core of both approaches is composed of feedback signals, scarcity of the item data impacts reducing directly the accuracy of these methods. Figure 61 supports hypothesis $H3$ by showing the low accuracy of these methods when the venue is very sparse, in the left bottom corner. In Figure 61a, the x-axis indicates check-ins frequency of the relevant venue normalized by the frequency of the most popular venue in the neighborhood (e.g., $x = 1.0$ indicates test cases where the relevant venue is the most popular in its neighborhood). Higher frequency indicates that the relevant item is more popular among its neighbors. The bars below the x-axis represent a distribution of the test data throughout the frequencies. Considering that the evaluated approaches do not handle item cold-start scenario, Figure 61b shows the progress of each algorithm in a magnified view of relevant items for extremely low check-ins frequency.

From Figure 61b, we first observe that some approaches are more sensitive to the number of venue check-ins. Memory-based approaches, such as ItemKNN and UserKNN, outperform the random approach when the feedback frequency reaches 0.02 and 0.04 respectively, which indicates that these algorithms can learn and improve the accuracy with very few check-ins for the relevant venue. More feedback is needed to make the WRMF and WBPRMF better than random approach, 0.06 and 0.07 respectively. Despite the fact that MP and BPRMF are the best approaches overall, they require more feedback to achieve a better recommendation, which makes them unsuitable for high item sparsity cases. In particular, these approaches require a minimum relative popularity of 0.08 of the target venue compared to other venues to outperform a random approach. Surprisingly Rank-GeoFM was the approach most affected by extreme item sparsity, achieving a better result than a random approach only after the relevant item reaches a relative popularity of 0.12.

Rank-GeoFM has the worst performance among all POI recommenders for test cases where the relevant POIs relative check-in frequency is lower than 0.14. In turn, it delivers the top performance for test cases with relevant POI frequency greater than 0.3. This might be caused by the loss function of Rank-GeoFM, which learns to rank positive examples higher than negative examples [Liu et al., 2017]. Thus, the more sparse the data is, the worse Rank-GeoFM learning will be, that is lower accuracy. In addition, experiments in the Singapore and São Paulo collections showed that despite the good performance of the Rank-GeoFM in general, its accuracy has improved only after the POI received at least 20 check-ins (Appendix C4 and C5).

We also observe that BPRMF obtains lower accuracy for high item sparsity than WBPRMF, and gets better only when the relevant item has a relative frequency above 0.18. To better understand the benefits of frequency weighting for matrix factorization, we compare the reciprocal rank (RR) of the test cases between WBPRMF and BPRMF. Figure 62 compares the difference between the RR of each test case, i.e., $\Delta RR = A_{score} - B_{score}$, where A is the WBPRMF Reciprocal Rank score and the B is the BPRMF Reciprocal Rank score. Figure 62a presents the ΔRR of all tests cases which shows that BPRMF has superior accuracy overall. Despite that, Figure 62b indicates that in high sparsity scenarios WBPRMF seems to be more effective by sampling items with probability proportional to their popularity. Therefore, when the relevant item has a relative frequency lower than 0.16, the WBPRMF is the best choice over BPRMF.



(a) $\rho >= 0$ (b) $\rho < 0.18$

Figure 62: Difference between the reciprocal rank of WBPRMF and BPRMF. ρ is the popularity of the relevant POI normalized by its set of candidates. The higher value, the more popular the relevant POI is in its neighborhood.

6.2.1.2 User Sparsity

Another important aspect that influences the accuracy of a POI recommender is the user visitation history. Check-ins represent the preferences of users, hence, scarcity of users' check-ins can cause a lower accuracy for all methods. To analyze the user sparsity impact over POI recommenders, we investigate how reliable and accurate these recommenders are through different users' history size. Like the previous analysis, we propose the following hypotheses:

$H4$. New users tend to visit popular POIs, whereas experienced users visit POIs of varying popularity.

To test hypothesis $H4$, we conduct an evaluation analyzing the number of check-ins per user. In this analysis, we split test cases according to the users history size in a range of 0 to 1000. We consider a cold-start users those with no previous check-ins; highly-sparse users those with less than 10 check-ins in training, and expert users those with more than 10 check-ins. If our hypothesis is supported, methods that leverage popularity should outperform other methods when users have up to 10 check-ins. According to Figure 63, users with fewer check-ins are more likely to visit popular POIs. Indeed, MP has the best accuracy for cold-start and highly-sparse users, but its advantage compared to other methods decreases as as more expert users are considered.

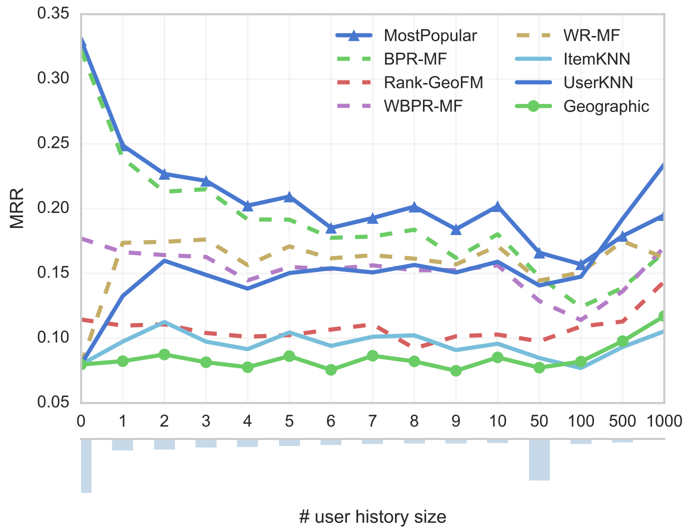
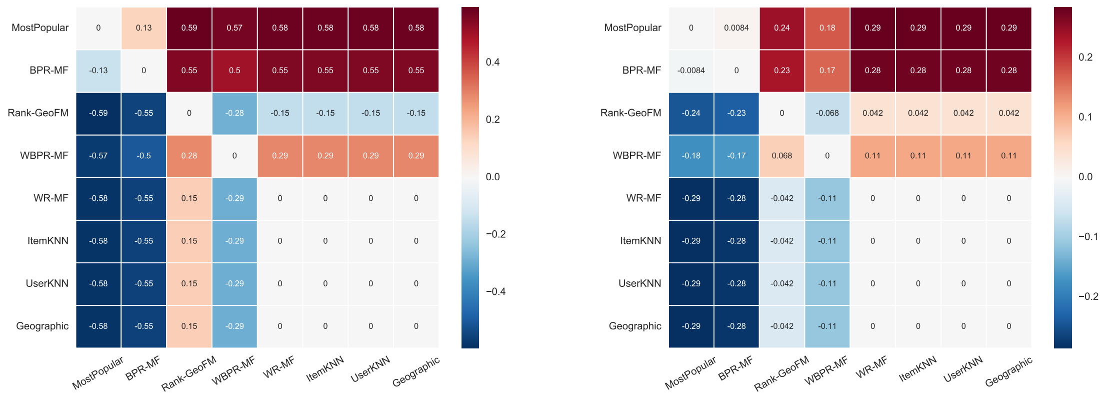


Figure 63: Accuracy of methods for users with different numbers of check-ins. Memory-based methods are represented by a line with no marker; Model-based methods are a dotted line; and Non-personalized methods are indicated by a line with markers.

In agreement with the previous analysis, we can observe a well-known problem from the collaborative filtering approaches that suffer from lack of user data. Due to the cold-start problem, collaborative filtering methods would fail to recommend venues, since they feedback to compute item similarities between the users. Figure 64a shows the win-loss comparing of pairs of algorithms, where the value in each cell represents the percentage of winning test cases by the row-algorithm over column-algorithm. The red color indicates that the row-algorithm scores higher than column-algorithm, otherwise blue ones indicate a lower score. Moreover, Figure 64b shows a comparison between the accuracy of the algorithms, but the cell label is the ΔMRR of the row-column algorithms. We can notice that both matrices have in the lower right corner a square of gray cells filled with zeros, which represent methods that could not improve or even recommend any different POI from the initial rank - i.e., Geographic approach. Aside Bayesian methods, the personalized algorithms that depend merely on users' feedback cannot deal with users cold-start problem.

The user cold-start analysis bring us two more interesting facts about the accuracy of collaborative recommenders. From Figure 63, we observe that (a) personalized algorithms do not vary significantly in terms of accuracy in non-cold-start scenarios. Despite the increasing size of user history, the accuracy hardly changes. Some methods are sensitive only after 100 check-ins and may experience changes in accuracy as demonstrated by UserKNN and WBPRMF. The second finding is that (b) the accuracy of the popularity-based approaches decreases with increasing number of user feedbacks. At the same 100 users' check-ins, the Most Popular method has an MRR 55% lower than in user cold-start scenario, and 36% lower compared to users with only one check-in. However, the MP approach is indifferent to the user sparsity. Thus, we assume that the loss of accuracy is due to the users' behavior and not the algorithm itself. Further supporting hypothesis $H4$, this suggests that inexperienced users prefer popular places, while expert users prefer to explore the long tail of unvisited places.



(a) Δ win-loss (in percentage) (b) Δ Mean Reciprocal Rank

Figure 64: Win-loss comparison between approaches when the user is cold-start. The red color indicates that the row-algorithm scores higher than column-algorithm, otherwise blue ones indicate a lower score. In Figure 64a, the value in each cell is the row-algorithm percentage of winning test cases over the column-algorithm. The cell value in Figure 64b is the difference between the MRRs of the row and columns algorithms.

In summary, we point out three main observations with user sparsity analysis. First, our hypothesis $H4$ is supported by Figure 63, showing that the Most Popular algorithm outperforms personalized approaches. Second, personalized algorithms are ineffective in cold-start scenarios, as they are unable to change the original ranking. Third, personalized algorithms are indifferent when the user is not cold-start. Last, the popularity-based approaches accuracy decrease with additional user feedback, which suggest that expert users might prefer unpopular places or visit new places instead of places they already knew.

6.2.1.3 System Cold-Start

After investigating the user and item cold-start scenarios individually, we further analyze considering the influence of system cold-start (i.e., no knowledge about the user and the item to be recommended). As the previous analysis showed in Section 6.2.1.2, collaborative POI recommenders underperform with limited prior information about users and POIs. In this section we aim to verify the following hypothesis:

*H*5. The overall popularity of the set of candidates negatively affects collaborative POI recommenders in the absence of prior information about the target user and the relevant POI.

Although our system cold-start scenarios represent only 3% of our test cases, this analysis may give an insight of the behavior of the POI recommenders and how accurately they perform in a region with limited data or for new systems.

Usually, three strategies can be applied to minimize the cold-start problem. First, Content-based filtering uses the metadata and complementary information about the POI (e.g., category, reviews, price range, etc.) to create recommendations. While the strategy is focused in the item cold-start scenario, a second strategy aims to recommend popular POIs to a cold-start user (known as popularity based approach). The third strategy consists in adopting a hybrid recommender, which can combine multiple strategies, by combining and leveraging the strengths of previous strategies. Since content-based and hybrid recommenders are outside the scope of our investigation, we direct our focus to the popularity-based strategy.

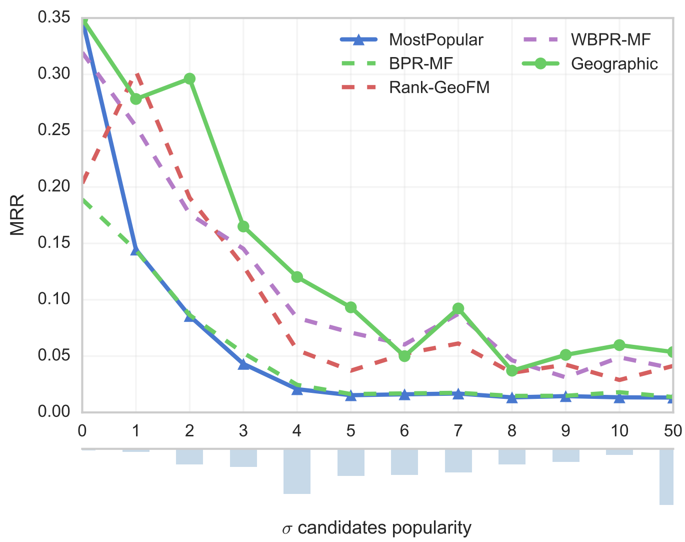


Figure 65: The standard deviation of the set of candidates popularity in system cold-start scenario.

Figure 65 measures how the popularity of the candidate POIs affects the accuracy of the methods. Bear in mind that in this analysis we aim to recommend a POI that has no check-in feedback in training to a user that also has no check-in either. The x-axis is the standard deviation of the candidate's popularity, and the y-axis is the MRR metric. The standard deviation helps analyze the distribution of popularity in the set of candidates and verify whether there is disparity between them. First, we noticed that all methods a lower accuracy than the random recommendation produced by the Geographic approach, which confirms that none of them is able to cope with the system cold-start problem. Second, accuracy decreases with increasing popularity of the set of candidates. Although the relevant POI is a cold-start item, the evaluated methods recommend POIs using the information learned from its check-ins, promoting candidates that have some historical data instead of cold-start POIs, which leads to a worse recommendation. This evidence supports hypothesis $H5$ by showing that the candidates' popularity intensifies the system cold-start problem since the algorithms promote popular candidates instead of those unvisited (yet relevant) POIs. Lastly, while the accuracy of the observed methods decreases as candidates popularity deviation increases from 0 to 1, the MRR of Rank-GeoFM improves in 49%. This might be caused by the geographical influence, where the Rank-GeoFM estimates a users interest in a POI based on its neighbors.

6.2.1.4 Non-Cold-Start

Non-cold-start test cases represent almost 58% of our test collection, which means that both the user and the relevant POI have some feedback. In Table 62 we note even though it is considered the state-of-the-art for POI Recommendation, Rank-GeoFM achieves a poor performance compared to others methods. On the other hand, Most Popular and BPRMF outperform all other methods by at least 10%. To further investigate this observation, we formulate the following hypothesis:

$H6$. Since the POI popularity is not global, the most popular place in the neighborhood becomes the best POI to recommend.

In order to examine hypothesis $H6$, we analyze some characteristics of the non-cold-start test cases as: the number of candidates per test case; and (b) the distribution of the item popularity, considering that the popularity distribution of POIs allows one to verify the existence of a concentration of popular POIs in our non-cold-start test cases. A small number of POIs in the set of candidates might contribute to a higher MRR accuracy of the algorithms. Although all methods benefit from a smaller candidate set, this analysis also will serve to exclude the chance of bias.

Figure 66 shows a distribution of the size of candidate sets containing an average of 188 candidates per test case. In the left corner of the distribution, we note a

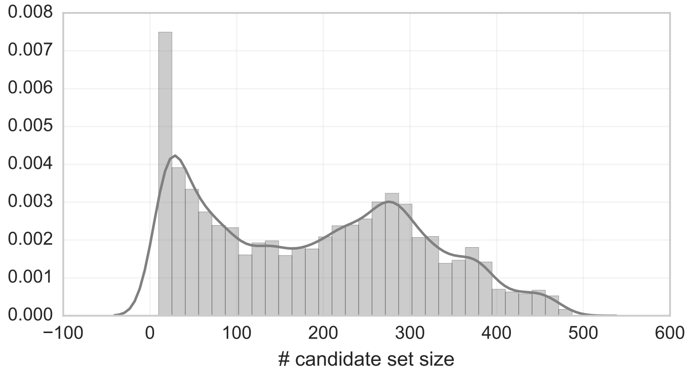


Figure 66: Distribution of the number of candidates per test case (only non-cold-start test cases).

peak near the value of 10 POIs per candidate set. To reduce the influence of smaller candidate sets, we restrict our analysis to candidate sets with at least 50 POIs. Thus, we remove 4,748 (20%) non-cold-start test cases. Even with this new constraint, the Most Popular approach remains the best method in this scenario.

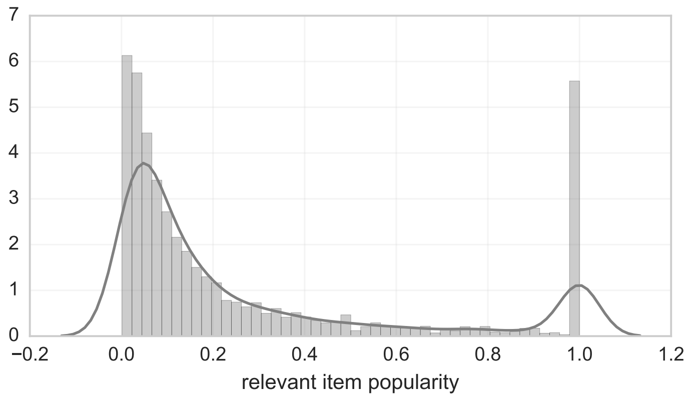


Figure 67: Distribution of the relevant item popularity with non-cold-start test cases. Item popularity is normalized and represent the relative popularity of the POI in the set of candidates.

In Figure 67 we inspect the distribution of the popularity of the relevant item. In the right side we observe a peak, which means a huge number of test cases with the relevant POI being very popular. Due to this evidence, our first thought is that popularity-based methods are unduly promoted. However, removing popular POIs and considering only test cases where the relevant items relative popularity is lower than 0.8 which represents 16,630 (72%) of the non-cold-start test cases, MP still outperforms all methods. Figure 68 shows that the MP is the best algorithm even when the relevant POI has a relative popularity lower than 0.5.

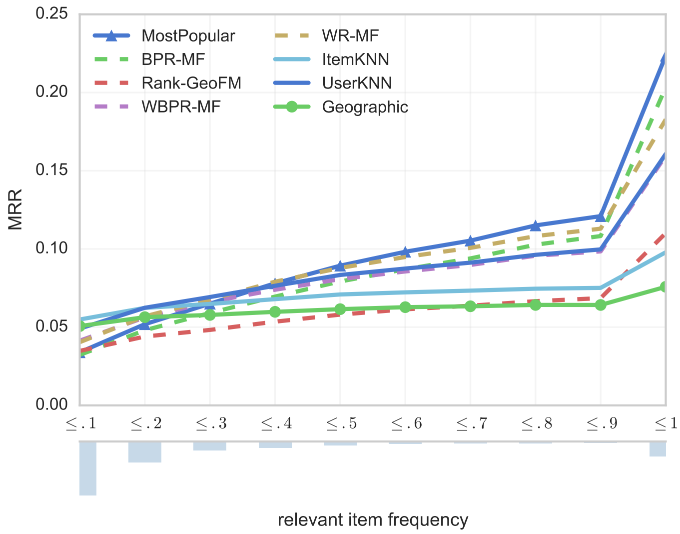


Figure 68: Relevant item frequency normalized by a set of candidates. Only non-cold-start test cases.

Despite popularity-based algorithms being demonstrably effective in several points, the distribution of check-ins is not entirely from popular items. Observing Figures 67 and 68, both left corners show that a substantial concentration of test cases with unpopular relevant POIs. More specifically there are 16,628 test cases where the relevant POI has a relative popularity below 0.3. Moreover, analyzing only these 72% of non-cold-start test cases, we find that either Most Popular or the other evaluated algorithms are not able to provide good recommendations of POIs with low frequency of check-ins. In light of these findings, the good accuracy of the Most Popular approach is due to 20% of test cases with highly popular relevant POIs. Despite the promising results of the Most Popular method, our mixed findings could not support hypothesis $H6$, and additional analysis is required.

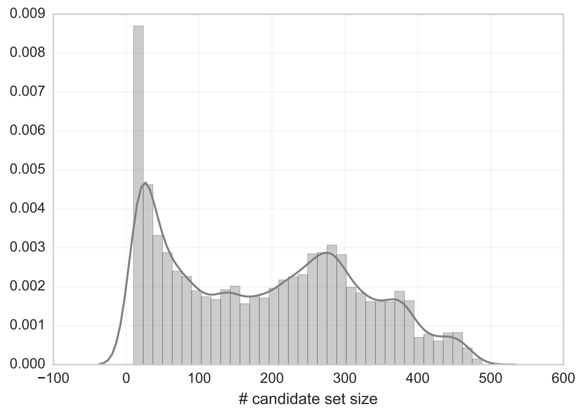
6.2.2 Neighborhood Density

Context-awareness is well-known for boosting the accuracy of recommender systems. In this analysis, we examine how contextual features related to the density of POIs in each neighborhood affect the accuracy of the POI recommenders and the previous analyses. Our interest in this study is due to the Nearby POI Recommendation task, which considers geography for constraining recommendations to a single neighborhood. The Nearby POI Recommendation task defines the concept of neighborhood as an area near the user's location. This area is bounded by a radius r from the user's location. Thus, the neighborhood definition leads us to the follow questions: (i) *What happens to* *the recommendation if we consider different range size?*; and as the area density might vary between neighborhoods with the same radiuses, (ii) *How do POI recommenders behave with different density of the neighborhoods?*. To address these questions, we pose the following three new hypothesis:

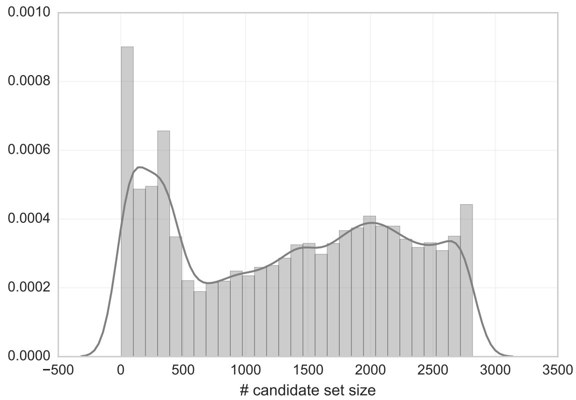
$H7$. Dense neighborhoods represent more POIs as candidates to the recommendation which increases the noise that POI recommenders must deal with it;

$H8$. Dense neighborhoods inhibit the recommendation of cold-start items;

$H9$. Cold-start items are more likely to be recommended when they have high scoring neighbors.



(a) r = 500 meters



(b) r = 1 kilometer

Figure 69: Distribution of the rank size using user's location in a range of 500 meters and 1 kilometer from the relevant POI.

In order to answer the first question, we conduct an additional experiment varying the neighborhood boundary. Figure 69 shows a distribution of the candidate set size using radiuses of 500 meters and 1 kilometer to generate the neighborhood. We can observe that increasing the exploratory area of the user - i.e., the radius of the neighborhood - more places will be available in the set of candidates hence more challenging it will be to suggest the relevant item. The average of candidate set size increased from 188 to 1335 candidate POIs just by changing the neighborhood radius from 500 meters to 1 kilometer. Observing Figure 610, we note a disparity in the accuracy of the algorithms using more POIs as candidates. In particular, increasing the radius to create the neighborhood may increase the difficulty for POI recommenders to suggest the relevant item at the top of the list. Therefore, using a dynamic range to build

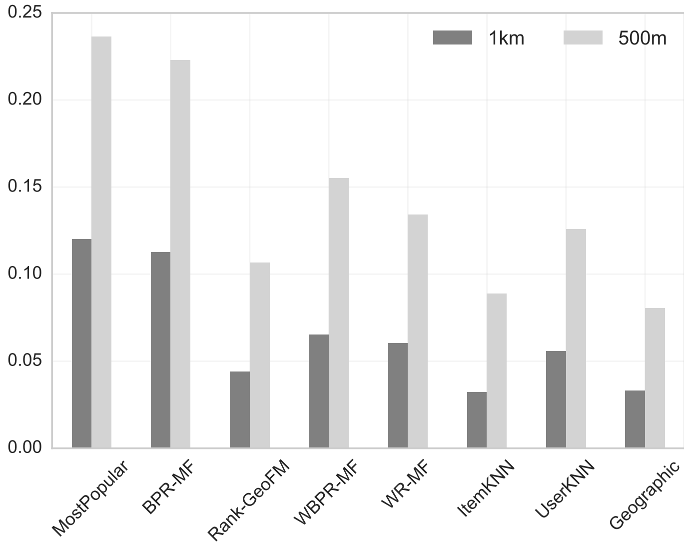
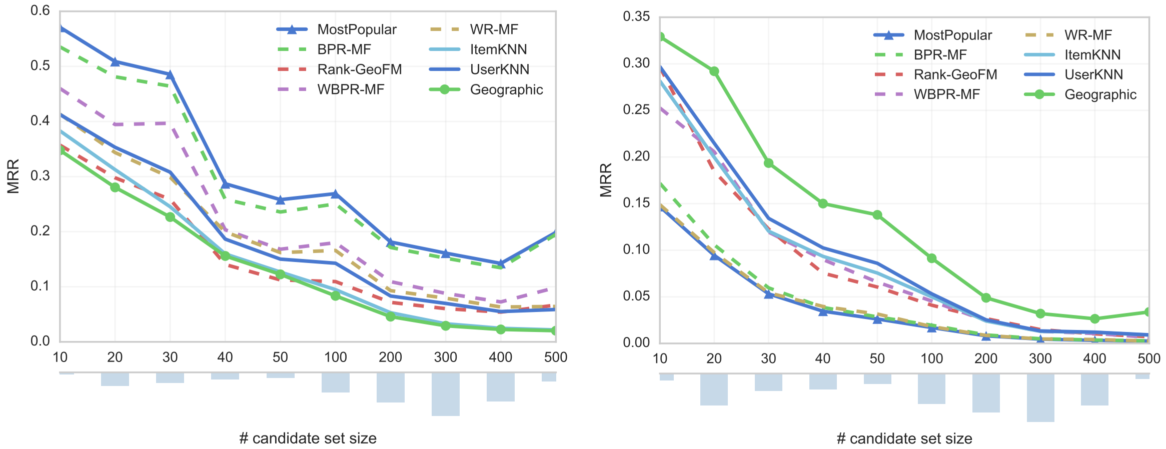


Figure 610: The accuracy of the algorithms using 500 meters and 1 kilometer in the radius size of the neighborhood.

neighborhoods might be beneficial when having different user behaviors, for instance, users who are willing to go further to visit a place.

While the previous analysis explored the neighborhood density by varying its radius to build the set of candidate POIs, we now assess the impact of density for a fixed radius of 500 meters. Figure 611 shows that the higher the number of candidates, the lower the accuracy of all recommenders. As we claimed in hypothesis $H7$, increasing the number of candidate POIs contributes to adding more noise to the recommendation, consequently hindering the accuracy of POI recommenders. Moreover, looking at Figures 611a and 611b, we see that high density makes it difficult to recommend POIs in general and exacerbate the problem to cold-start items, hence, supporting $H8$.

In addition to the global density of a neighborhood, another important aspect that might impact the effectiveness of POI recommenders is the influence that the immediate vicinity of each candidate POI exerts on the POI. As mentioned earlier, Rank-GeoFM models a score that measures a user's likelihood of liking a POI because of its immediate neighbors. To cope with this geographical influence, we introduce the *POI neighborhood score*, which is the sum of the popularity of its K nearest POIs. In this fashion, we simulate Rank-GeoFM's geographical score to investigate the importance of popularity of neighboring items for the accuracy of POI recommenders. In particular, our hypothesis $H9$ aims to test whether a relevant POI having popular neighbors contributes to achieving a better recommendation when the POI is cold-start. Figure 612 shows the methods accuracy only for item cold-start test cases, where the x-axis is the neighborhood score of the relevant POI. Like Rank-GeoFM,



(a) All test cases (b) Only item cold-start test cases

Figure 611: The accuracy of the algorithms using the neighborhood densities. A radius $r = 500$ was used to build the set of candidate POIs.

our neighborhood score was calculated using the $K = 100$ nearest POIs. We note that neighborhood score attenuates the item cold-start problem, however, the problem remains since the algorithms are still worse than a random method. Notwithstanding the fact that a random approach is the best method in our study for item cold-start problem, there is an accuracy improvement of the algorithms when popular neighbors surround a cold-start POI, thus, it supports our hypothesis $H9$.

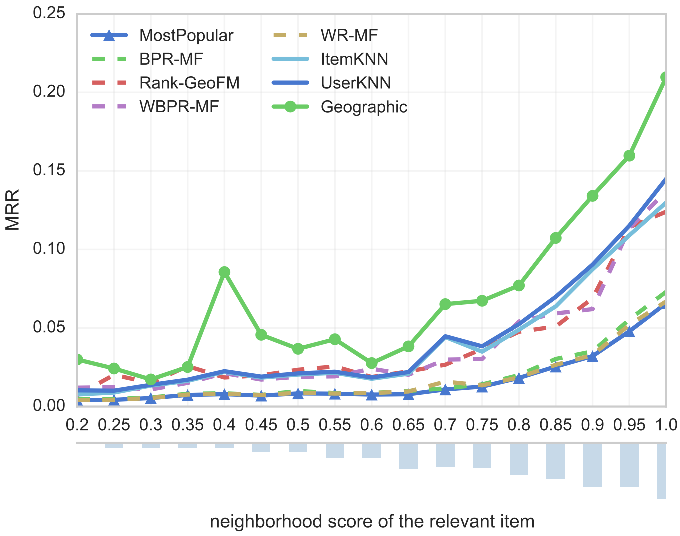


Figure 612: The accuracy of the algorithms by the neighborhood score in item cold-start scenario. The neighborhood score is the sum of the popularity of the k closest POIs of the target POI. $k = 100$.

6.3 Pruned Training

Location-based pruning on POI recommendation is an interesting approach that improves the efficiency of the algorithms as well as models a more realistic scenario for POI recommendation. In addition to the experiments conducted in Section 6.1, in this section we compare the effectiveness-efficiency impact of the Nearby POI recommendation task during training. To this end, we introduce the Nearby Rank-GeoFM, a variation of the Rank-GeoFM algorithm which leverages the location-based pruning on the training step (details in Appendix B).

As aforementioned, Rank-GeoFM is the state-of-the-art recommender for POI recommendation. It makes a pairwise comparison of POIs measuring incompatibility between the expected ranking and the ranking produced by a factorization model. Rank-GeoFM iterates through all the user-POI check-ins and updates the latent factors until the model converges. In each iteration, the method performs a pairwise comparison, sampling one POI from the POIs $\mathcal{V}$ until it finds an incorrectly-ranked POI to update the latent factors.

Likewise, Nearby Rank-GeoFM also iterates through all the user-POI check-ins, learning the latent factors of the model. However, Nearby Rank-GeoFM performs the sampling from the set of candidate POIs $\mathbf{C}_{l_u} \subset \mathcal{V}$ instead of the entire set $\mathcal{V}$. This customization adds the location-based pruning characteristic of the Nearby POI Recommendation task, thus considering only near POIs from the user's location during the training of the algorithm. In accordance with our Nearby POI Recommendation task, this implementation restricts the state-of-the-art recommender to work using data in the user's vicinity.

In this section, we analyze $RQ1$ regarding the impact of pruning the training data. To this end, we pose the following hypotheses:

$H10$. The effectiveness of the Nearby Rank-GeoFM remains the same. Pruning the training reduces noise of out-ranged POIs.

$H11$. The efficiency of the Nearby Rank-GeoFM will be improved proportionally to the pruning size.

We conduct two analyses to check the hypotheses above. The first analysis attempts to validate $H10$, by comparing the effectiveness of the Rank-GeoFM method and its pruned variation. To compare the methods, we evaluate the training of both algorithms under the same conditions using the New York City collection with 500m and 1 km. The effectiveness results for both methods are presented in Table 63. Results show that despite Rank-GeoFM's pruning, the effectiveness of the methods remains unchanged[1]. These results indicate that the location-based pruning contributes to removing out-ranged POIs from the training and does not affect the learned latent factors. Although these preliminary results support hypothesis $H10$, we believe further analyses with other recommenders are needed to strengthen this assertion.

Table 63: A effectiveness comparison pruning the training process of the Rank-GeoFM. Nearby Rank-GeoFM is the modified algorithm to pruned the training considering only the nearby POIs as candidates. MRR is measured in two different neighborhood and user's locations ranges: 1 kilometer and 500 meters.

	Mean Reciprocal Rank	
method	**1 km**	**500 m**
Rank-GeoFM	0.0440	0.1066
Nearby Rank-GeoFM	0.0432	0.1063

The second analysis is related to the efficiency of the methods. In particular, we investigate whether employing the location-based pruning to the training collection drives the Nearby Rank-GeoFM to faster learning. To test hypothesis $H11$, we examine the elapsed time for each iteration of the methods. A single iteration is composed by $\mathcal{D}$ iterations where $\mathcal{D}$ is the amount of user-POI check-ins in the training data. We set both methods to iterate 1000 times through all the user-POIs check-ins ($1000 \times \mathcal{D}$). The total time elapsed for the training of Rank-GeoFM was 7h 32min whereas the pruned versions using 1km and 500 meters took 6h 29min and 3h 47min respectively. Therefore, Nearby POI Rank-GeoFM reduces the training time by 50% as shown in Table 64.

Table 64: A comparison of iteration time by pruning training data of the Rank-GeoFM in New York City collection.

method	**total**	**min. iter.**	**max.**	**avg.**
Rank-GeoFM	7h 32min	21.20s	38.70s	27.20s
Nearby Rank-GeoFM (1 km)	6h 29min	21.07s	30.31s	23.39s
Nearby Rank-GeoFM (500 m)	3h 47min	11.56s	21.33s	13.66s

The Nearby Rank-GeoFM adaptation responsible for improving the method's efficiency is implemented in lines 6∼8 of Algorithm 1. Pruning the training data assures that the pairwise comparisons performed by Rank-GeoFM drops from $k < |\mathcal{V}|$ to $k < |\mathbf{C}_{l_u}| \ll |\mathcal{V}|$, where k denotes the number of sampling trials before obtaining an

[1]The tiny difference in the MRR accuracy of the methods may be due to the random initialization of Rank-GeoFM matrices.

incorrectly-ranked POI to update the latent factors. Figure 613 shows that the Rank-GeoFM iterations take longer when the training reaches a stable phase, in contrast, the Nearby Rank-GeoFM iterations remain stable since the number of samplings to find a ranking incompatibility will not rise over time. Supporting hypothesis $H11$, this preliminary analysis of pruning the training data showed an improvement in the efficiency of Nearby Rank-GeoFM and demonstrated a faster learning compared to non-pruned Rank-GeoFM. However, further studies are needed to analyze other methods and collections.

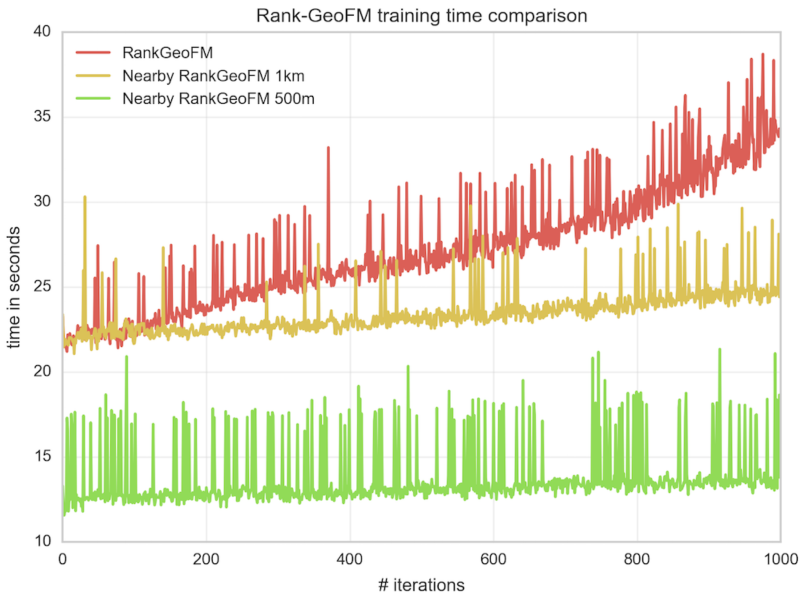


Figure 613: Time comparison between the Rank-GeoFM and Nearby Rank-GeoFM over 1000 iterations. Axis-x shows the time elapsed in seconds to complete the iteration.

6.4 Summary

In Chapter 4 we proposed the Nearby POI recommendation task having the geographical context built into the POI recommendation task. Chapter 5 detailed the experimental setup modeling a more realistic scenario for POI recommenders by using the user's location and geographical context of the venues to evaluate the performance of the POI recommenders. In this chapter, we evaluated the performance of eight POI recommenders using our proposed Nearby POI recommendation task. We conducted several experiments analyzing the impact of the location-based pruning strategy both in training as well as at prediction time.

In Section 6.1 we presented the impact of location-based pruning for POI recommendation on test cases. To answer the first research question, we evaluated the POI recommenders comparing the results of the traditional recommendation task (not pruning test cases) and the Nearby POI recommendation task (pruning test cases). This analysis showed that by adding geographic context to the task can impact on the effectiveness of some POI recommenders. Moreover, adopting a location-based pruning reduces the computational cost, hence, improving the efficiency of the POI recommenders.

In Section 6.2 we performed a breakdown analysis comparing the effectiveness of the POI recommenders for different sparsity conditions. First, in Section 6.2.1.1 we analyzed the POI recommenders accuracy over the item-sparsity. We showed how collaborative filtering recommenders are severely harmed by the scarcity of POI check-ins. Also, we observed that the poor accuracy of Rank-GeoFM is due to the extreme data sparsity of the collection. We also presented an analysis showing in which item sparsity condition the POI recommender starts to give better recommendations.

In Section 6.2.1.2 we assessed the methods regarding user sparsity conditions. In this section we noted that personalized algorithms which depend merely on users' feedback (e.g., UserKNN, ItemKNN and WRMF) cannot deal with users cold-start problem. In contrast, we observed that popularity-based methods performed better recommending POIs to users with few check-ins. Furthermore, we found that experienced users visit POIs of varying popularity whereas new users tend visit popular POIs.

In Section 6.2.1.3, we analyzed the accuracy of the methods on a system cold-start scenario. We showed that all methods accuracy were lower compared to a random recommendation produced by the Geographic approach. Hence, none of the POI recommenders evaluated in this dissertation are able to cope with the system cold-start problem. In addition, we demonstrated that the candidates' popularity intensifies the system cold-start problem.

In Section 6.2.1.4 we conducted experiments analyzing the accuracy of the POI recommenders for an active user (i.e., who have already checked any POI). We performed an extensive analysis throughout the check-ins in order to find out why the non-personalized MP performed better in this scenario. We believed that since the POI popularity is no longer global (i.e., considering all POIs in the target city as candidates for POI recommendation), the most popular POIs in the neighborhoods would be better recommendations. However, removing test cases where the target POI is very popular, we found that either MP or other evaluated recommenders could not provide good recommendations. Therefore, the good accuracy of the MP is due to test cases with highly popular relevant POIs.

Section 6.2.2 examined the impact of how the neighborhood (i.e., the set of candidate POIs) affects the accuracy of the POI recommenders. We compare two different range sizes (i.e., 500m and 1km) to build the neighborhoods. We showed that the higher the number of candidates in the neighborhood, the lower the accuracy of all recommenders. Also, we discussed how higher density jeopardizes the recommendation of cold-start items. Lastly, we analyzed the user's likelihood of liking a POI because of its immediate neighbors, and we note that neighborhood score attenuates the item cold-start problem.

In addition, in Section 6.3 we suggested an adaptation of the Rank-GeoFM method regarding the use of location-based pruning in learning the users' preferences in training. The results showed an improvement in the efficiency of our proposed Nearby Rank-GeoFM adaptation without significant losses in effectiveness.

Chapter 7

Conclusions and Future Work

POI recommendations received extensive research attention in the last few years, and many approaches have been proposed. In most of these studies, the developed POI recommenders exploit the geographical influence to learn the user's preferences. Indeed, researchers have a common agreement that geography information is a key feature of POI recommendation. In addition, researchers have argued about impact of the geography for user's mobility and showed that users usually tend to visit POIs in their vicinity [Bao et al., 2015]. However, previous studies overlook this information by considering all POIs in the target city as candidates for recommendation. In this dissertation, we proposed the Nearby POI recommendation task which leverages a location-based pruning strategy to restrict the recommendations to POIs in the user's vicinity. We showed that adding geography into POI recommendation can improve the efficiency of POI recommenders (by removing the cost of prediction for out-ranged POIs) without losses in their effectiveness. Moreover, we presented a breakdown analysis evaluating eight state-of-the-art recommender methods through different levels of check-in data sparsity as well as for neighborhoods of various densities.

In the remainder of this chapter, Section 7.1 revisits the main contributions of this work, and we summarize the conclusions drawn from the investigations conducted in this dissertation. Section 7.2 presents several ideas for future research, directly stemming from the results of this dissertation.

7.1 Summary of Contributions

We summarize the main contributions of this dissertation as follows.

A large scale test collection for context-aware nearby POI recommendation. In Chapter 3, we presented a test collection for POI recommendation which includes check-ins from Foursquare, spread over one year in three large cities in the Americas. The collection was enriched with temporal, weather, social and geographical contexts. In contrast to existing collections, we assess the effectiveness of a POI recommender at ranking candidate POIs constrained by the user's location. Thus, several geography-constrained test cases were included in the collection.

We propose to build geography into POI recommendation. In Chapter 4, we formalized the nearby POI recommendation task to exploit the geographic context given the user's location. The task aims to explore the fact that users tend to visit nearby places through a location-based pruning for POI recommendation. Section 4.1 explained the strategy employed to simulate the user's location based on the assumption that the user is near the target POI. Section 4.2 described the location-based pruning strategy to construct the users' neighborhoods and thus build geography into POI recommendation. Furthermore, in Chapter 6 we performed an evaluation of several state-of-the-art recommenders using the location-based pruning strategy. Our results showed that efficiency improvements can be attained both at learning as well as at prediction time without significant losses in effectiveness.

A deeper breakdown analyses of several POI recommenders. In Chapter 6, we compared 8 state-of-the-art POI recommenders on our test collection for context-aware nearby POI recommendation. Through a breakdown analysis, we further demonstrated the effectiveness-efficiency trade-offs incurred for different levels of user-POI sparsity as well as for neighborhoods of various densities.

Our findings suggest that inexperienced users prefer popular places whereas expert users seek for new places thus exploring more the long-tail. Also, the evaluated methods suffer from the item cold-start problem and are not able to promote new POIs for the user. Another observation is that by restricting candidate POIs to the users neighborhood, non-personalized popularity-based methods will promote local popular places.

Finally, as we believe that geography should be a fundamental user constraint in the POI recommendation, the nearby POI recommendation decoupled geography from other effects on POI recommendation. Our proposed evaluation methodology diminishes any trivial favoring that might exist for geography-aware POI recommenders, as demonstrated by Rank-GeoFM being penalized in comparison other methods.

7.2 Directions for Future Research

In this section, we discuss possible directions for future research stemming from the research of this dissertation.

In Chapter 4, we introduced the nearby POI recommendation as a task builds geography into POI recommendation. While we focused on a static pruning technique, with a fixed radius for all users, a promising direction is to build POI recommenders using dynamic pruning to find optimal radius per user or location. Another approach to consider is the construction of adaptive neighborhoods by distance, POI categories and different densities.

Despite the fact that we conducted a breakdown analysis of POI recommenders in Chapter 6, we believe that an additional analysis involving more context-aware POI recommenders is necessary. Also, a breakdown analysis considering other collections would collaborate with a better understanding of the algorithms independent of geography.

Another direction is analyzing the impact of diversity in POI recommendation. To be our best knowledge, the diversity of POI recommendations has seen limited research thus far [Chen et al., 2015; Han and Yamana, 2017]. We conducted initial experiments employing the xQuAD framework [Santos et al., 2010] to promote diversity on POI recommendations. We obtained promising results increasing the coverage of different categories for POI recommendation without losses in accuracy.

Finally, the proposed evaluation methodology enables further analyses such as a clearer assessment of the role of contextual factors other than geography on a user's decision to check in at a POI. In addition, an analysis using additional pruning radiuses could lead to interesting findings.

Appendix A

Nearby POI Collection

Table A1: The list of fields on a tweet

field	description
tweet id	identifier for tweet
text	tweet text
swarm url	swarm check-in url
place	indicates where tweet is associated
coordinates	geographic location of tweet
language	language of the tweet
created	UTC time when tweet was created
timestamp	timestamp when tweet was created
user id	identifier for user
screen_name	alias that this user identifies themselves @something
user	The name of the user
user verified	indicates that the user has a verified account

Table A2: The list of climate features

feature	description
temperature	the air temperature in degrees Celsius
apparent temperature	"feels like" temperature in degrees Celsius
precipitation probability	between 0 and 1
icon	a machine-readable text summary (e.g., clear-day, cloudy, fog, partly-cloudy-night)
summary	a human-readable text summary (e.g., snow, heavy rain and breeze, mostly cloudy, etc.)
cloud cover	the percentage of sky occluded by clouds
humidity	the relative humidity

Table A3: The full list of available features in Nearby POI collection

type	feature	
user	identifier	gender
POI	identifier name main category	sub-categories tags
check-in	identifier user user's coordinates POI temperature apparent temperature temperature icon precipitation probability cloud cover	utc time check-in timestamp set of candidate POIs weekday is weekend season daytime summary of the temperature humidity
tips	identifier lang	text timestamp
social	user identifier POI	friend identifier check-in time

Table A4: Most commonly collections for POI Recommendation. The following marks indicate that the collection ✔ possesses such information.

	period	location	# users	# POIs	# check-ins	# tips	# reviews	category	temporal	geographical	weather	social
Foursquare 1 †	Aug. 2010 to Jul. 2011	Singapore	2,321	5,596	194,108	-	-		✔	✔		
Foursquare 2 ‡	Dec. 2009 to Jun. 2013	California	4,163	121,142	483,813	-	-	✔		✔		
Foursquare 3 ⋆	Apr. 2012 to Sep. 2013	415 cities (77 countries)	266,909	3,680,126	33,278,683	-	-	✔	✔	✔		
Gowalla †	Feb. 2009 to Oct. 2010	California and Nevada	10,162	24,250	456,988	-	-		✔	✔		✔
Nearby POI	Aug. 2015 to Jul. 2016	3 cities (2 countries)	49,601	118,46	1,386,63	462,641	-	✔	✔	✔	✔	✔

† Yuan et al. [2013, 2014]

‡ Li et al. [2016]

⋆ Yang et al. [2015, 2016]

Appendix B

Nearby Rank-GeoFM

The Nearby Rank-GeoFM is a variation of the ranking based geographical factorization method (Rank-GeoFM). Like Rank-GeoFM, Nearby Rank-GeoFM learns the ranking models based on a pairwise comparison between the POIs, measuring the incompatibility between the generated rankings. To determine the optimal values for latent matrices, the Rank-GeoFM needs to iterate through all the user-POI check-ins, updating the latent factors until model converges. Also, for each user-POI, the Rank-GeoFM makes a pairwise comparison, sampling a POI ℓ' from the POIs $\mathcal{V}$ until finds an incorrectly-ranked POI to update the latent factors. On the other hand, to fulfill the Nearby POI recommendation task concept, we need to change the Rank-GeoFM sampling method to consider only the near POIs from the user's location. Nearby Rank-GeoFM incorporates the pruning characteristic in the training step just changing the sample ℓ' from the entire set of POIs $\mathcal{V}$ to the $\mathbf{C_{l_u}}$. In Algorithm 1 we can observe this change in lines 6∼8. In addition, we provide the set of candidate POIs $\mathbf{C_{l_u}}$ from the user's vicinity combined to the user-POI check-in (line 3).

B.1 Complexity Analysis

Based on Algorithm 1, the complexity of making a pairwise comparison is $O(rD\bar{s})$, where r is the number of iterations until convergence, D is the number of check-ins for training (i.e., non-zero entries in the user-POI check-in matrix) and $\bar{s}$ is the average of the number of trials until finding an incompatible example. For each pairwise comparison, Rank-GeoFM computes the recommendation score using Equation 5.2. Hence, the complexity of computing a recommendation score is $O(Kk)$, where K is the number of dimensions of the latent space, and k is the number of neighbor POIs. Thus, the training complexity of Rank-GeoFM is $O(rD\bar{s}Kk)$. Furthermore, for most iterations, Rank-GeoFM updates the latent matrices with a cost of $O(Kk)$. In addition to the training step, the recommendation step only considers Equation 5.2 complexity $O(Kk)$. The main difference of the complexity of Nearby Rank-GeoFM is reducing the number of trials $\bar{s}$ to find an incompatible example. Since the Nearby Rank-GeoFM explores a smaller set of candidate POIs, the sampling $\bar{s}$ will be smaller.

Algorithm 1 Nearby Rank-GeoFM

input: check-ins training data $\mathcal{D}$
output: model parameters $\Theta = \left\{\mathbf{U}^{(1)}, \mathbf{L}^{(1)}, \mathbf{U}^{(2)}\right\}$

Initialize Θ with Normal distribution $\mathcal{N}(0, 0.01)$
repeat
    for $(u, \ell, \mathbf{C}_{l_u}) \in \mathcal{D}$ **do**
        $n \leftarrow 0$
        compute score $y_{u\ell}$
        repeat ▷ sampling through the candidates C
            sample a POI $\ell' \in (C \subset \mathcal{L})$
            compute score $y_{u\ell'}$
            $n \leftarrow n + 1$
        until RANKING INCOMPATIBILITY == 1 or $n > |C|$
        if RANKING INCOMPATIBILITY == 1 **then**
            UPDATE LATENT FACTORS
        end if
    end for
until *convergence*
return $\Theta = \left\{\mathbf{U}^{(1)}, \mathbf{L}^{(1)}, \mathbf{U}^{(2)}\right\}$

Appendix C

Additional Results

Table C1: The Nearby POI recommendation task in different collections.

	New York City	**Singapore**	**São Paulo**
MP	**0.2365**	0.3153	**0.3321**
BPRMF	0.2228	0.3109	0.2747
WBPRMF	0.1552	0.1863	0.2293
WRMF	0.1341	0.3068	0.2350
Rank-GeoFM	0.1066	**0.3280**	0.1783
UserKNN	0.1259	0.2749	0.2283
ItemKNN	0.0889	0.2324	0.1915
Geographic	0.0806	0.0887	0.2037

Table C2: The effectiveness of recommenders for Nearby POI Recommendation task in the Singapore collection.

	Mean Reciprocal Rank				
	(100%) **total**	(0,5%) **item cs**	(0%) **user cs**	(0%) **cs**	(99,5%) **non-cs**
MP	0.3153	0.0213	-	-	0.3170
BPRMF	0.3109	0.0214	-	-	0.3125
Rank-GeoFM	**0.3280**	0.0280	-	-	**0.3297**
WBPRMF	0.1863	0.0574	-	-	0.1870
WRMF	0.3068	0.0231	-	-	0.3084
UserCF	0.2749	0.0351	-	-	0.2763
ItemCF	0.2324	0.0360	-	-	0.2335
Geographic	0.0886	**0.0733**	-	-	0.0887

Table C3: The effectiveness of recommenders for Nearby POI Recommendation task in the São Paulo collection.

	Mean Reciprocal Rank				
	(100%) **total**	(21,4%) **item cs**	(18,9%) **user cs**	(9,1%) **cs**	(50,6%) **non-cs**
MP	**0.3321**	0.0544	**0.4840**	0.0544	**0.4392**
BPRMF	0.2747	0.0557	0.2956	0.0997	0.4077
Rank-GeoFM	0.2293	0.1049	0.1974	**0.1159**	0.2137
WBPRMF	0.2350	0.1100	0.2511	0.1137	0.2937
WRMF	0.1783	0.0696	0.2564	0.1090	0.3228
UserCF	0.2283	0.1199	0.2564	0.1090	0.2875
ItemCF	0.1915	0.1099	0.2564	0.1090	0.2186
Geographic	0.2037	**0.1295**	0.2564	0.1090	0.2343

Table C4: A performance comparison using items with different numbers of check-ins in the Singapore collection.

	≤ 0	≤ 1	≤ 2	≤ 5	≤ 10	≤ 20	≤ 50	≤ 100	∞
MP	0.0213	0.0229	0.0250	0.0304	0.0415	0.0589	0.1045	0.1577	0.3153
BPRMF	0.0214	0.0239	0.0275	0.0339	0.0442	0.0607	0.1032	0.1544	0.3109
Rank-GeoFM	0.0280	0.0291	0.0301	0.0386	0.0549	0.0833	**0.1368**	**0.1848**	**0.3280**
WBPRMF	0.0574	0.0563	0.0524	0.0647	0.0798	**0.0967**	0.1179	0.1365	0.1863
WRMF	0.0231	0.0271	0.0314	0.0418	0.0508	0.0682	0.1108	0.1593	0.3068
UserKNN	0.0351	0.0383	0.0429	0.0552	0.0630	0.0741	0.1004	0.1380	0.2749
ItemKNN	0.0360	**0.0858**	**0.1109**	**0.0963**	0.0852	0.0814	0.0906	0.1150	0.2324
Geographic	**0.0733**	0.0808	0.0936	0.0956	**0.0984**	0.0957	0.0936	0.0947	0.0886

Table C5: A performance comparison using items with different numbers of check-ins in the São Paulo collection.

	≤ 0	≤ 1	≤ 2	≤ 5	≤ 10	≤ 20	≤ 50	≤ 100	∞
MP	0.0549	0.0713	0.0877	0.1388	**0.1769**	**0.2152**	**0.2779**	**0.3162**	**0.3321**
BPRMF	0.1080	0.1026	0.1003	0.1040	0.1098	0.1257	0.1545	0.1731	0.1783
Rank-GeoFM	0.0692	0.0835	0.0976	0.1292	0.1551	0.1825	0.2306	0.2603	0.2747
WBPRMF	0.1083	0.1107	0.1101	0.1175	0.1329	0.1590	0.1984	0.2193	0.2293
WRMF	0.0792	0.0944	0.1023	0.1224	0.1392	0.1632	0.2025	0.2257	0.2350
UserKNN	0.1153	0.1305	0.1366	0.1493	0.1591	0.1712	0.1995	0.2170	0.2283
ItemKNN	0.1082	0.1254	0.1324	0.1445	0.1524	0.1624	0.1785	0.1863	0.1915
Geographic	**0.1223**	**0.1361**	0.1397	**0.1497**	0.1569	0.1661	0.1864	0.1973	0.2037

Bibliography

Bannur, S. and Alonso, O. (2014). Analyzing temporal characteristics of check-in data. In *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*, WWW Companion '14, pages 827--832, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Bao, J., Zheng, Y., Wilkie, D., and Mokbel, M. (2015). Recommendations in location-based social networks: a survey. *GeoInformatica*, 19(3):525--565. ISSN 1573-7624.

Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 43--52, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Burke, R. (2000). Knowledge-based recommender systems. 69.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331--370. ISSN 0924-1868.

Cambazoglu, B. B. and Baeza-Yates, R. (2011). *Scalability Challenges in Web Search Engines*, pages 27--50. Springer Berlin Heidelberg, Berlin, Heidelberg.

Carmel, D., Cohen, D., Fagin, R., Farchi, E., Herscovici, M., Maarek, Y. S., and Soffer, A. (2001). Static index pruning for information retrieval systems. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 43--50, New York, NY, USA. ACM.

Chen, X., Zeng, Y., Cong, G., Qin, S., Xiang, Y., and Dai, Y. (2015). On information coverage for location category based point-of-interest recommendation. In *AAAI*, pages 37--43.

Cheng, Z., Caverlee, J., Lee, K., and Sui, D. (2011). Exploring millions of footprints in location sharing services.

Cho, E., Myers, S. A., and Leskovec, J. (2011). Friendship and mobility: User movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1082--1090. ACM.

Dasgupta, S. (2016). Fast collaborative filtering from implicit feedback with provable guarantees. In Durrant, R. J. and Kim, K.-E., editors, *Proceedings of The 8th Asian Conference on Machine Learning*, volume 63 of *Proceedings of Machine Learning Research*, pages 206--221, The University of Waikato, Hamilton, New Zealand. PMLR.

Ferreira, A. P. G., Silva, T. H., and Loureiro, A. A. F. (2015). Beyond sights: Large scale study of tourists' behavior using foursquare data. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 1117–1124. ISSN .

Gao, H., Tang, J., Hu, X., and Liu, H. (2013). Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 93--100, New York, NY, USA. ACM.

Gorakala, S. K. and Usuelli, M. (2015). *Building a Recommendation System with R*. Packt Publishing. ISBN 1783554495, 9781783554492.

Guo, L., Jiang, H., Wang, X., and Liu, F. (2017). Learning to recommend point-of-interest with the weighted bayesian personalized ranking method in lbsns. *Information*, 8(1). ISSN 2078-2489.

Han, J. and Yamana, H. (2017). Geographical diversification in poi recommendation: Toward improved coverage on interested areas. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, pages 224--228, New York, NY, USA. ACM.

Hu, L., Sun, A., and Liu, Y. (2014). Your neighbors affect your ratings: On geographical neighborhood influence to rating prediction. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 345--354, New York, NY, USA. ACM.

Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 263--272, Washington, DC, USA. IEEE Computer Society.

Kang, Z., Peng, C., and Cheng, Q. (2016). Top-n recommender system via matrix completion. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 179--184. AAAI Press.

Khatwani, S. and Chandak, M. (2016). Building personalized and non personalized recommendation systems. In *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, pages 623–628. ISSN .

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30--37. ISSN 0018-9162.

Li, H., Ge, Y., Hong, R., and Zhu, H. (2016). Point-of-interest recommendations: Learning potential check-ins from friends. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 975--984, New York, NY, USA. ACM.

Li, X., Cong, G., Li, X.-L., Pham, T.-A. N., and Krishnaswamy, S. (2015). Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 433--442, New York, NY, USA. ACM.

Lian, D., Zhao, C., Xie, X., Sun, G., Chen, E., and Rui, Y. (2014). Geomf: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 831--840, New York, NY, USA. ACM.

Liao, Y., Lam, W., Jameel, S., Schockaert, S., and Xie, X. (2016). Who wants to join me?: Companion recommendation in location based social networks. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, ICTIR '16, pages 271--280, New York, NY, USA. ACM.

Liu, B., Fu, Y., Yao, Z., and Xiong, H. (2013). Learning geographical preferences for point-of-interest recommendation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 1043--1051, New York, NY, USA. ACM.

Liu, Y., Pham, T.-A. N., Cong, G., and Yuan, Q. (2017). An experimental evaluation of point-of-interest recommendation in location-based social networks. *Proc. VLDB Endow.*, 10(10):1010--1021. ISSN 2150-8097.

Liu, Y., Wei, W., Sun, A., and Miao, C. (2014). Exploiting geographical neighborhood characteristics for location recommendation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 739--748, New York, NY, USA. ACM.

Long, X., Jin, L., and Joshi, J. (2013). Towards understanding traveler behavior in location-based social networks. In *2013 IEEE Global Communications Conference (GLOBECOM)*, pages 3182–3187. ISSN 1930-529X.

Meehan, K., Lunney, T., Curran, K., and McCaughey, A. (2013). Context-aware intelligent recommendation system for tourism. In *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 328–331, San Diego, CA, USA. IEEE, IEEE. ISSN .

Moraes, F., Vasconcelos, M., Prado, P., Dalip, D., Almeida, J. M., and Gonçalves, M. (2013). Polarity detection of foursquare tips. In *Proceedings of the 5th International Conference on Social Informatics - Volume 8238*, SocInfo 2013, pages 153--162, New York, NY, USA. Springer-Verlag New York, Inc.

Noulas, A., Scellato, S., Lathia, N., and Mascolo, C. (2012). A random walk around the city: New venue recommendation in location-based social networks. In *Proceedings of the 2012 ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust*, SOCIALCOM-PASSAT '12, pages 144--153, Washington, DC, USA. IEEE Computer Society.

Noulas, A., Scellato, S., Mascolo, C., and Pontil, M. (2011). An empirical study of geographic user activity patterns in foursquare.

Pan, R., Zhou, Y., Cao, B., Liu, N. N., Lukose, R., Scholz, M., and Yang, Q. (2008). One-class collaborative filtering. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 502--511, Washington, DC, USA. IEEE Computer Society.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452--461, Arlington, Virginia, United States. AUAI Press.

Resnick, P. and Varian, H. R. (1997). Recommender systems. *Commun. ACM*, 40(3):56--58. ISSN 0001-0782.

Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. (2010). *Recommender Systems Handbook*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition. ISBN 0387858199, 9780387858197.

Santos, R. L., Macdonald, C., and Ounis, I. (2010). Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 881--890, New York, NY, USA. ACM.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285--295, New York, NY, USA. ACM.

Tobler, W. R. (1970). A computer movie simulating urban growth in the detroit region. *Economic Geography*, 46:234–240. ISSN 00130095, 19448287.

Trattner, C., Oberegger, A., Eberhard, L., Parra, D., and Marinho, L. (2016). Understanding the impact of weather for POI recommendations. In *Proceedings of the Workshop on Recommenders in Tourism co-located with 10th ACM Conference on Recommender Systems (RecSys 2016), Boston, MA, USA, September 15, 2016.*, pages 16--23, Boston, MA, USA. ACM.

Yang, D., Zhang, D., Chen, L., and Qu, B. (2015). Nationtelescope: Monitoring and visualizing large-scale collective behavior in lbsns. *Journal of Network and Computer Applications*, 55:170--180.

Yang, D., Zhang, D., and Qu, B. (2016). Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Trans. Intell. Syst. Technol.*, 7(3):30:1--30:23. ISSN 2157-6904.

Yang, D., Zhang, D., Yu, Z., and Wang, Z. (2013). A sentiment-enhanced personalized location recommendation system. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, HT '13, pages 119--128, New York, NY, USA. ACM.

Ye, J., Zhu, Z., and Cheng, H. (2013). *What's Your Next Move: User Activity Prediction in Location-based Social Networks*, pages 171–179.

Ye, M., Yin, P., Lee, W.-C., and Lee, D.-L. (2011). Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 325--334, New York, NY, USA. ACM.

Yu, Y. and Chen, X. (2015). A survey of point-of-interest recommendation in location-based social networks.

Yuan, Q., Cong, G., Ma, Z., Sun, A., and Thalmann, N. M. (2013). Time-aware point-of-interest recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 363--372, New York, NY, USA. ACM.

Yuan, Q., Cong, G., and Sun, A. (2014). Graph-based point-of-interest recommendation with geographical and temporal influences. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 659--668, New York, NY, USA. ACM.

Zhao, S., King, I., and Lyu, M. R. (2016). A survey of point-of-interest recommendation in location-based social networks. *CoRR*, abs/1607.00647.

Zheng, Y. (2012). Tutorial on location-based social networks. WWW 2012.