2-21-2019

Context-Aware Personalized Point-of-Interest Recommendation System

Ramesh Raj Baral
rbara012@fiu.edu

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

CONTEXT-AWARE PERSONALIZED POINT-OF-INTEREST RECOMMENDATION SYSTEM

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Ramesh Baral

2019

To: Dean John L. Volakis
College of Engineering and Computing

This dissertation, written by Ramesh Baral, and entitled Context-aware Personalized Point-of-Interest Recommendation System, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Shu-Ching Chen

Ning Xie

Leonardo Bobadilla

Debra VanderMeer

Sundaraja Sitharama Iyengar, Major Professor

Date of Defense: February 21, 2019

The dissertation of Ramesh Baral is approved.

Dean John L. Volakis
College of Engineering and Computing

Andres G. Gil
Vice President for Research and Economic Development
and Dean of the University Graduate School

Florida International University, 2019

DEDICATION

To my father Dinanath, mother Durga, brothers Prakash and Bishnu, wife Kavya, and daughter Krisha.

ACKNOWLEDGMENTS

First of all, I would like to thank Professor Sundaraja Sitharama Iyengar for his kind supervision during my Ph.D. Without his encouragement and guidance, I would not have been able to complete my Ph.D., and this dissertation would not have existed. I would also like to thank my former advisor Professor Tao Li for his motivation and supervision during my preliminary research. I would also like to thank Professor Shu-ching Chen, Professor Ning Xie, Professor Leonardo Bobadilla, and Professor Debra VanderMeer for being my doctoral committee. They have provided many valuable suggestions and insights for my dissertation. I extend my warmest thanks to my supervisors at National Center for Atmospheric Research (NCAR), who gave me help and support during my summer internships. I would also like to thank all of my coauthors and lab mates with whom I have collaborated. It was my great honor to work with them. I am also thankful to our school's administrative and advising staffs Professor Jason Liu, Olga Carbonell, Vanessa Cornwall, Carlos Cabrera, Ariana Taglioretti, and Luis Rivera. As some parts of this thesis were originally published in ACM RecSys 2016, IEE IRI 2016, Journal of Data Mining and Knowledge Discovery 2017, ACM UMAP 2018, ACM RecSys 2018, and ACM SIGSPATIAL 2018, I am also thankful to all the reviewers, editors, and conference organizers who provided valuable feedback on my research. Finally, I would like to thank my parents and family for their motivation and support.

ABSTRACT OF THE DISSERTATION

CONTEXT-AWARE PERSONALIZED POINT-OF-INTEREST RECOMMENDATION SYSTEM

by

Ramesh Baral

Florida International University, 2019

Miami, Florida

Professor Sundaraja Sitharama Iyengar, Major Professor

The increasing volume of information has created overwhelming challenges to extract the relevant items manually. Fortunately, the online systems, such as e-commerce (e.g., Amazon[1]) and the location-based social networks (LBSNs) (e.g., Facebook[2]) among many others have the ability to track end users' browsing and consumption experiences. Such explicit experiences (e.g., ratings, likes/dislikes, etc.) and many implicit contexts (e.g., demographic, social, spatial, temporal, and categorical, etc.) are useful in preference elicitation and recommendation. As an emerging branch of information filtering, the recommendation systems are already popular in many domains, such as movies (e.g., YouTube[3]), music (e.g., Pandora[4]), and Point-of-Interest (POI) (e.g., Yelp[5]).

The POI domain has many contextual challenges (e.g., spatial (preferences to a near place), social (e.g., friend's influence), temporal (e.g., popularity at certain time), categorical (similar preferences to places with same category), locality of POI, etc.) that can be crucial for an efficient recommendation. The

[1] www.amazon.com

[2] www.facebook.com

[3] www.youtube.com

[4] www.pandora.com

[5] www.yelp.com

user reviews shared across different social networks provide granularity in users' consumption experience. From the data mining and machine learning perspective, following three research directions were identified and considered relevant to an efficient context-aware POI recommendation, (1) incorporation of major contexts into a single model and a detailed analysis of the impact of those contexts, (2) exploitation of user activity and location influence to model hierarchical preferences, and (3) exploitation of user reviews to formulate the aspect opinion relation and to generate explanation for recommendation. This dissertation presents different machine learning and data mining-based solutions to address the above-mentioned research problems, including, (1) recommendation models inspired from contextualized ranking and matrix factorization that incorporate the major contexts and help in analysis of their importance, (2) hierarchical and matrix-factorization models that formulate users' activity and POI influences on different localities that model hierarchical preferences and generate individual and sequence recommendations, and (3) graphical models inspired from natural language processing and neural networks to generate recommendations augmented with aspect-based explanations and interpretation of the generated recommendation.

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

CHAPTER 1

INTRODUCTION

The evolution of the World Wide Web (WWW) and smart-phone technologies have played a key role in the revolution of our daily life. The service providers (e.g., e-commerce systems, such as Amazon[1], eBay[2], etc.) post their product for sale and the end users share their consumption experience via ratings, tags, likes-dislikes, short tips, and reviews. Often, the end users scan through the items they need, observe and analyze others' consumption experience (e.g., ratings and reviews), and finally select the item that matches their preferences. The increasing volume of information has made it overwhelmingly difficult to filter and extract the information manually and locate the items relevant to user preferences.

Fortunately, the service providers, such as e-commerce (e.g., Amazon[1], eBay[2], etc.), location-based social networks (LBSNs) (e.g., Facebook[3], Foursquare[4], etc.), and many others have the ability to store end users' browsing, consumption history, and several demographic attributes, correlate them with other users' consumption behavior and with the items in their repositories. These systems can exploit such information for users' preference elicitation. Some of the systems even allow users to share explicit preferences and consumption experiences in terms of star-ratings (e.g., rating of 5 is for best experience and rating of 1 for worst experience), text reviews (e.g., users can write their experience in free text or can fill out a template), likes-dislikes (e.g., users can click on the like or dislike icon to share their experience), and tags (e.g.,

[1] www.amazon.com

[2] www.ebay.com

[3] www.facebook.com

[4] www.foursquare.com

users can tag the items using their keywords) or tips (e.g., users can suggest few words to recommend or not to recommend the items). The explicit preferences (e.g., ratings, likes-dislikes, etc.) and implicit predictors or contexts (e.g., songs listened, web pages accessed, social relations, spatial attributes, demographic attributes, consumption times, item categories, aspect term preferences, etc.) are the key factors used to filter relevant information and elicit the potential interests of end users. This concept of information filtering is the key idea behind the evolution of recommendation system that focuses on the exploitation of explicit and implicit preference information to predict the potential preferences of end users.

Conceptualized in early 90s [GNOT92, RIS$^+$94, SM95], the recommendation systems are already popular in many domains, such as music (e.g., Pandora[5]), movies (e.g., YouTube[6]), books (e.g., Amazon[1]), social tags (e.g., Facebook[3]), experts[7], social and professional networks (e.g., LinkedIn[8]), dating partners[9], point-of-interests (e.g., Yelp[10]), and products in general. Each and every recommendation system adopt the technique that suits the features relevant to them.

The classical recommendation systems fall under three core techniques: (1) Collaborative filtering (CF): It builds a model from two major techniques: (a) memory-based: it uses user-based CF (a user's past behavior (items purchased, selected, rated) is correlated with other users' behavior to find similarity pref-

[5] www.pandora.com

[6] www.youtube.com

[7] www.aminer.org

[8] www.linkedin.com

[9] www.match.com

[10] www.yelp.com

erences between users, and the items consumed by users with similar preferences are used to predict the potential preferences) or item-based CF (exploits item-item similarity matrix and infers the likelihood of an item based on its similarity to already consumed items), and (b) model-based: it uses available data to build and train models which are then used to predict the preferences, (2) Content-based filtering (CBF): It utilizes the features of an item (often termed as a seed item) and recommends items that have similar properties to the items previously consumed by users, and (3) Hybrid approach: It combines both CF and CBF (e.g., a system that uses viewing and searching trends of similar users (i.e., CF) and also the items that have features common to the items that are rated highly by a user (i.e., CBF)). Although the classical models are very popular, they have some limitations. The CF technique suffers from the cold-start problem (unpredictability due to lack of user preference information), scalability (computationally expensive when there are many user-item entries), and data sparsity (regardless of the number of items, users rate or review only few items and there is no explicit preference information for rest of the items). The CBF tends to infer items similar to the seed item and lacks variability in the recommended items.

Existing studies [AT11, ZTZX14, GZC+09] have shown that the cold-start and data sparsity problems can be partially solved by context-aware recommendation systems (CARS) which are the recommenders that incorporate different contexts (e.g., time of the day, current location of user, social status of user, time budget, and similar factors that have (in)direct influence on end users' preferences) to leverage the quality of recommendations by recommending the items that are contextually relevant and match user preferences (e.g., recommending bar at evening and night is more relevant than recommending it in the morning). The personalized recommendation systems incorporate individual user preferences because the preferences vary on items and contexts (e.g., some users might be interested in cheap items regardless of the distance but other users might prefer near items regardless of the price, some users might prefer good service and may be willing to pay high but others might focus on cheap items, etc.).

Our study focuses on one of the emerging branches of recommendation, known as Point-of-Interest (POI) recommendation which exploits the check-in experience (e.g., POI visits), multimedia contents (e.g., review text), and different contexts (e.g., check-in time) shared on LBSNs. Unlike the general product recommenders, the POI domain has special features and contextual challenges; for instance, the check-in frequencies vary across different users and places, resulting in the *sparsity* of the user-location rating matrix. A check-in activity is influenced by many contextual challenges (e.g., *spatial* (preferences to a near place), *social* (influence of the social tie (e.g., friendship)), *temporal* (influence of temporal check-in pattern (e.g., the popularity of *bars* is in the evenings and nights)), *categorical* (similar preferences to places with the same category), the *utility* of a POI regardless of the distance or cost, popularity of POI (due to social or other impact), *dynamic mobility* (trend to visit new places), *promotions and coupons*, *popularity* of a POI, *locality* of POI, *current time*, previous check-in *category*, *time budget*, *price*, etc.) that can be crucial for an efficient recommendation.

From the perspective of machine learning and data mining, we identified three useful research directions relevant to efficient context-aware POI recommendations.

1. Multi-context POI Recommendation: The user check-ins are influenced by many factors, such as the current time, location category, social relation (e.g., friends or family), spatial (distance to POI), previous check-in, locality, etc. Exploitation of these factors can be crucial for an efficient recommendation; for instance, exploiting the current *time* of a day can

be used to filter out the relevant POIs (e.g., better to recommend bars at evening or night, some places might operate on specific hours or seasons, etc.), the *distance* factor can be used to filter out distant places, the target check-ins can vary with *social* factor (e.g., different preferences for family, friends, alone, etc.), unavailability of a potential preferred place can be diverted to place with similar *category* (e.g., visiting a nearby cafe if the nearest coffee shop is closed), and so on. Existing studies have shown significant improvement in recommendation quality by incorporating such contexts (e.g., geographical [YYLL11, BZM12, WTM13, FYL13, HE13, ZC15], temporal [YCM+13, JSW+12, WTM13, HJE13], social [YYLL11, CYKL12, FYL13, WTM13, ZC15], categorical [BZM12, HSL14, RW13, LLAM13, ZC15], sentiment [YZYW13], popularity [RW13, LLAM13]. The existing studies have not incorporated all the major contexts in a single recommendation model. A detailed analysis of the impact of major factors (social, spatial, temporal, and the categorical) and their incorporation for POI recommendation is a viable research direction to explore.

2. Locality-aware POI Recommendation: The check-in behavior is contextually dynamic and varies with the context, locality of visit, co-consumers, etc. Generally, the check-ins within a geographical region are cluttered around some centers (e.g., popular POIs) which influence the check-ins on nearby POIs (e.g., the *Empire State building* has some influence on the check-ins of nearby POIs). Similarly, users have some check-in or activity trend (e.g., activity of a user varies by locality, item type, etc.). The POI influence and user activity can be mapped to a joint latent space to derive their latent features and can be exploited for POI recommendation [LZX+14, GAN15]. The contextual extension of such latent factors is still an interesting direction to explore. The variation of contextual

preference also implies cluttering of different preference trends in each region. Another interesting research direction is to efficiently aggregate the preference trends to model the locality preferences of users.

3. Explainable POI Recommendation: The end users share their consumption experience via reviews. The reviews are one of the important features of LBSNs and are helpful to elaborate opinions and share the extent of consumption experience in terms of relevant factors of interest or aspects (e.g., “Despite the high *price* of the camera, the photo *quality* was bad”). Though some of the review-aware recommenders exist, most of them are less transparent and non-interpretable (as they conceal the reason behind recommendation). Some of the studies [SNM08, TM07, VSR09, TM12, GJG14] have already claimed the persuasiveness of explanation for real-world systems. To the best of our knowledge, none of the existing studies have explored review-aware explainable POI recommendation. The explainable recommenders in other domains have coupled the influence of all aspects. As the aspects have some influence among themselves, it is better to model them individually. For instance, a place that is good in “Price”category might be opposite in “Service”. A user who just cares about the “Price”aspect might ignore some “Service”related problems. An interesting research direction can be to separate the influence of aspects based on the order of aspect preference and use the aspects to generate explanation for recommendation (e.g., why is a POI recommended to a user?).

This dissertation addressed the research topics outlined above. Concretely, it focused on designing and developing data-driven solutions for contextual POI recommendation, including: (1) Exploitation of contexts for personalized POI recommendation, (2) Efficient modeling of location influence, user activity, and locality preference for contextual POI recommendation, and (3) Extraction of aspect terms and aspect categories from review text to model explainable and interpretable recommendation.

1.1 Problem Statement

This research focused on the concrete problems from each of the aforementioned directions, and presented the corresponding solutions. Specifically, the following research problems were studied: (1) *What will be the role of major contexts (e.g., spatial, temporal, categorical, and social) in POI recommendation and what will be the impact of incorporation of all these contexts in POI recommendation?* (2) *How do we incorporate the location influence, user activity, and locality preference for POI recommendation?*, and (3) *How do we extract aspect-based preferences from review text to model them for an explainable POI recommendation?*

1.2 Contributions

This dissertation addressed the research topics outlined above. Concretely, it focused on designing and developing data mining solutions to model the contextual POI recommendations that incorporate the user activity, location influence, and aspect-based explanation for the generated recommendation.

1.2.1 Contextual Point-of-Interest recommendation

The POI domain has many contexts that can have direct or indirect influence on the check-in behavior of users. Careful selection of contexts can significantly impact the efficiency of recommendation. In the study [BL16, BL17], we define and analyze the fusion of different major contexts. The preference of a user u to a location l at a time t is influenced by the check-in history of the user at the time t. For instance, if a user's check-in history has frequent check-ins in Starbucks coffee shop at 2 P.M., then it is more likely that she will visit a coffee shop at that time in the future. This temporal aspect should be considered while recommending some coffee (or relevant category) shops to her. If that coffee shop is inaccessible, the user might not be surprised if a nearby cafe is recommended. Such an affinity of the time and location category has motivated us to incorporate them in the POI rankings. We represent the check-in history as a graph where every location is termed as a node and the bag of ⟨users, time⟩ tuple is considered as its attribute. The location-location edges exist if they have same category or are within a threshold distance. This categorical and spatial sensitive model is inspired from the Topic-Sensitive PageRank [Hav02] and incorporates the categorical, social, spatial, and temporal contexts to rank the nodes. A personalized ranking relation is defined to model the preference of a user to a POI. The evaluation on two real-world datasets (Weeplaces and Gowalla) [LLAM13] using precision, recall, and F-Score metrics demonstrates the efficiency of the proposed model.

1.2.2 User activity and location influence on Point-of-Interest recommendation

We represent the locations as sequential grids of equal area, ensuring each grid with some check-ins. The user's influence area or activity area is defined as the region that depicts high possibility of the appearance of the user. The POI influence area is defined as the popularity of a POI within a grid. Our study [BWLC16] is influenced by a Non-negative matrix factorization to derive user and POI latent feature matrices, which are supplemented with the users' and POIs' influence on the grids and are then contextually exploited to generate efficient recommendation. The evaluation on two real-world datasets (Weeplaces and Gowalla) [LLAM13] using precision, recall, and F-Score metrics demonstrates the efficiency of our proposed model. In the study [BILZ18], we model the locality-based preference of user as a hierarchical structure and present a hierarchy aggregation technique to formulate the aggregated preferences of users and the locality trends. The extensive evaluation on two real-world datasets (Weeplaces and Gowalla) [LLAM13] using pair F-score, diversity, displacement, and NDCG metrics demonstrates significant performance gain of proposed model over baseline models and relevant studies.

1.2.3 Aspect-based explanation of Point-of-Interest recommendation

Most of the existing recommendation systems are not interpretable because they do not provide any explanation for the generated recommendation. An explanation of the recommendation is essential to persuade end uesrs and hence to maintain recommendation quality and usability. In the study [BZIL18], we formulate three different techniques to model user reviews to generate explainable POI recommendation. The evaluation of our proposed model on three real-world datasets (Yelp, TripAdvisor, and AirBnb) demonstrates its efficiency over several baselines and relevant studies.

1.3 Summary and Roadmap

The aforementioned research problems are organized and presented as follows: Chapter 2 presents a multi-context model for POI recommendation and also extensively analyzes the role of different contexts on POI recommendation. Chapter 3 presents a model that incorporates the user activity, location influence, and locality-based preference to generate POI recommendation. It also presents a model that formulates locality-based user preferences as preference hierarchy and presents a technique to aggregate the preference hierarchies. Chapter 4 presents a model that extracts aspects or features from the review text and formulates it to generate explanation or interpretation for POI recommendation. Chapter 5 presents some potential future directions and concludes the dissertation.

CHAPTER 2

MULTI-CONTEXT POINT-OF-INTEREST RECOMMENDATION

The evolution of the World Wide Web (WWW) and the smart-phone technologies have played a key role in the revolution of our daily life. The location-based social networks (LBSN) have emerged and facilitated the users to share the check-in information and multimedia contents. The Point-of-Interest (POI) recommendation system uses check-in information to predict the most potential check-in locations. The different attributes of check-in information, for instance, *geographical distance*, *category* and *temporal popularity* of a POI, and *temporal check-in trends* and *social (friendship)* information of a user play a crucial role in an efficient recommendation.

In this chapter, we present a fused recommendation model termed MAPS (**M**ulti **A**spect Personalized **P**OI Recommender **S**ystem) which fuses the categorical, temporal, social and spatial contexts into a single model. The major contributions of this research are: (i) it formulates the recommendation problem as a graph of location nodes with constraints on the category and the distance contexts (i.e. the edge between two locations is constrained by a threshold distance and the category of the locations), (ii) it proposes a multi-context fused POI recommendation model, and (iii) it extensively evaluates the proposed model with two real-world data sets.

2.1 Introduction

The LBSNs, such as Facebook[1], Foursquare[2], Gowalla[3], and so forth have facilitated users to share their check-in information relevant to places of interest.

[1] www.facebook.com

[2] www.foursquare.com

[3] www.gowalla.com

Such check-in information has been the subject of interest to predict the POIs that are most likely to be visited in the future. Albeit, the generic recommendation concept has been used for POI domain (for instance, the Collaborative Filtering (CF) [LSEM12], Content Based Filtering [YSC$^+$13], and Hybrid approaches [YZYW13]), its special contexts have motivated the community towards more sophisticated approaches for better results.

The frequency of check-ins varies across different users and places, resulting in the *sparsity* of the *user-location* frequency matrix in comparison to the *user-item* rating matrix in the generic systems. The check-in preference to a near place introduces the *spatial context* (the distance to a POI). Though the *social context* encourages to incorporate the social tie (*e.g., friendship*), it costs the challenge from the unreliability of check-in information diffusion, which is also supported by the findings from existing studies [YYL10] "only $\sim 96\%$ of people share $< 10\%$ of the commonly visited places and $\sim 87\%$ of people share nothing at all". The *temporal context* depicts the temporal check-in pattern. For instance, the popularity of the *bars* is in the evenings and nights. Many other relevant factors, such as (i) the utility of a POI, regardless of the distance, cost, (ii) the popularity of the POI (due to social or other impact), and (iii) the dynamic mobility of a user (trend to visit new places) exist. Although the problem is well explored [BWLC16, JSW$^+$12, WTM13, YYLL11, YCM$^+$13, ZC15], incorporation of all the major contexts (*the social, spatial, temporal, and categorical*) into a single model is barely explored.

2.2 Related Research

In this section, we group the relevant studies according to the context they incorporated into their research model.

2.2.1 Social context

Chao et al. [CML11] claimed that around 10% - 30% of human movements can be socially influenced. They found that the influence of friendship on user's mobility could be around 61% and the influence of mobility on new friendship could be around 24%. They used the Gaussian distribution with time of a day as a parameter to model the probability distribution over the latent states (work and home place) for a user. Contrary to the claim of Ye et al. [YYL10] ($\sim$ 96% of people share $<$ 10% of the commonly visited places and $\sim$ 87% of people share nothing at all), Gao et al. [GTL12] assumed that people share their check-in activities among friends. Their model used the Hierarchical Pitman-Yor (HPY) language model to represent the check-in pattern of a user and has shown effective results. Although these models exploited the social contexts, they did not focus on the temporal context of check-in activities.

2.2.2 Temporal context

Jin et al. [JSW$^+$12] proposed a graph-based model where the following/follower relation was realized as directed edge between user nodes. The nodes were ranked using the topic-sensitive PageRank [Hav02]. Though they incorporated the temporal context, other major contexts, such as geographical, categorical, and the social contexts were not explored. Yuan et al. [YCM$^+$13] incorporated the spatial and temporal contexts. The prediction of a check-in to a location was defined in terms of aggregate of visits count on that location across all the users. The check-in time constraint was introduced for the temporal similarity measure. Though the model incorporated spatial context by considering its impact on the check-in trend, it did not define the social and categorical contexts for recommendation. Though the matrix factorization model from

Gao et al. [GTHL13] achieved exciting results by incorporating the temporal contexts, however, other contexts were left unexplored.

2.2.3 Categorical context

Liu et al. [LLAM13] introduced the dependency of user's check-in behavior with her current location, and the implicit POI category preference based on the categorical patterns on check-in data. They used K-means clustering algorithm to group the users with similar check-in category and frequency values, and similar check-in time. The HITS [GKR98] based model from Bao et al. [BZM12] addressed the users' preferences and their social opinions. The users' location history was categorized according to the types (for instance, shopping, restaurants, etc.). A user-location matrix was used to identify the local experts who have the higher affinity towards a POI category, and the experts' social opinions were used for recommendation. Their model also did not address the temporal context.

2.2.4 Spatial context

The First Law of Geography from Tobler [Tob70] which states "*everything is related to everything else, but near things are more related than the distant things*" is relevant to POI recommendation as well. Ye et al. [YYL10] incorporated the check-in willingness factor [Tob70], social, and spatial contexts. The spatial influence was modeled by using Bayesian CF approach. The social context was incorporated by considering a user's friends' check-in behavior rather than finding similarities with all the users in the dataset. Liu et al. [LFYX13] used the geographical probabilistic factor analysis framework that focused on multiple factors, such as, the user check-in count, geographical influence on POI selection, user mobility nature, and so forth. They modeled the users' mobility behavior by using multinomial distribution over latent regions and different activity regions. The temporal context remained unexplored in their model as well.

2.2.5 Other fused models

Wang et al. [WTM13] defined a heterogeneous graph with user and location nodes and computed the nodes' rank. The unobserved places which have the highest rank and within a threshold distance (e.g., from user's house) were recommended. Yin et al. [YSC$^+$13] exploited the POIs' content information (for instance, item tags or category keywords) to link the content-similar spatial items. Liu et al. [LX13] incorporated the POIs' content into users' and POIs' profile and utilized the context-aware information through probabilistic matrix factorization. Hu et al. [HE13] used topic modeling to exploit the spatial and textual contexts of user posts. Cheng et al. [CYLK13] considered the users' movement constraint and proposed a successive personalized POI recommendation model using matrix factorization method which embedded the personalized Markov chains and the localized regions.

Wang et al. [WYC$^+$15] used both the users' personal interests and the preference of crowd (with same role, e.g., tourist or local) in the target region along with the co-occurrence pattern of spatial items and the content (for instance, the tags and category keywords) of those spatial items. The probabilistic generative model from Yin et al. [YZS$^+$15] exploited the geographical, temporal, word-of-mouth, and semantic effect. Xie et al. [XYW$^+$16] used the geographical, temporal, and semantic contexts in their heterogeneous graph embedding model that was based on the time decay method and was claimed to be an efficient predictor for the user's latest preferences. Lian et al. [LZX$^+$14] exploited matrix factorization to incorporate users activity area and POIs influence area and used the spatial clustering of users and POIs. Liu et al. [LXP+15] used a geographical probabilistic factor model for POI recommendation. Liu et al. [LLL+16] exploited the user interests and their evolving sequential preferences with temporal interval assessment. Hu et al. [HSL14] exploited the impact of geographical neighborhood of a place on its rating. Wang et al. [WWT+17] used the visual correlation between the places and the images posted by users. A recent study from Stepan et al. [SMDM16] incorporated the spatial, temporal and the social context in their recommendation model. None of these models fused all the major contexts to generate personalized POI recommendations.

2.3 Methodology

The PageRank [PBMW99] graph ranking model used the number and quality of the links to a web page to estimate its importance. Its extension Topic-Sensitive PageRank [Hav02] model introduced some bias to the PageRank vector. It incorporated the set of influential or representative (*or additional context relevant attributes*) topics to address the importance of particular topics. For a given query, it identified the most closely associated/contextual topics and such relevant topic-sensitive (biased) vectors were used to rank the documents satisfying the query. The convergence of PageRank is assured only if the graph is strongly connected and aperiodic [MR10]. This becomes true if we add a damping constant $(1-\alpha)$ to the rank propagation which improves the quality of PageRank not only by limiting the effect of the rank sinks [BMPW98], but also by assuring the convergence to a unique rank vector [Hav02].

Our model MAPS is influenced by Topic-Sensitive PageRank and the representative topics are spatial and categorical contexts of the LBSN. The rank of a location (l) in the context of a user (u) and a time (t) is influenced by the check-in history of user (u) at the time (t). For instance, if a user's check-in history has frequent check-ins in Starbucks coffee shop at 2 pm, then it is more likely that she will visit a coffee shop at that time in future. This temporal context should be taken care while recommending some coffee (or relevant category) shops to her. If that coffee shop is inaccessible, the user might not be surprised if a nearby cafe is recommended. Such a dual affinity of time and location category has motivated us to incorporate the categorical and temporal bias in the POI rankings.

Given two candidate POIs, suggesting the near one is more relevant [Tob70]. If the check-in history of a user depicts that the check-ins were made within some distance of other check-ins, then introducing the distance constraint might give better recommendation. MAPS uses such check-in trends to incorporate the spatial bias in the location ranking.

In MAPS, every location is represented as a node of a graph and the bag of ⟨user, time⟩ tuple is considered as an attribute of the location node. The location-location edges exist if they have the same category or are located within some threshold distance. It uses the categorical and the spatial bias in its context sensitive ranking model. The terms used in this paper are defined in Table 2.1. The categorical sensitive PageRank for MAPS is defined as:

$$\Pi^{c}_{t_1,t_2}(l) = \alpha * \beta_{t_1,t_2}(l) + (1-\alpha) * \sum_{(l'.cat=l.cat)} \Pi^{c}_{t_1,t_2}(l'), \tag{2.1}$$

where $\beta_{t_1,t_2}(l)$ is the categoric sensitive factor, defined as:

$$\beta_{t_1,t_2}(l) = \tau_1 * \frac{\sum\limits_{u \in U} \mid V_{u,t_1,t_2}(l) \mid}{\sum\limits_{u \in U, l.cat=l'.cat} \mid V_{u,t_1,t_2}(l') \mid} + \tau_2 * \frac{\sum\limits_{u \in U, l.cat=l'.cat} \mid V_{u,t_1,t_2}(l') \mid}{\sum\limits_{p \in L, u \in U} \mid V_{u,t_1,t_2}(p) \mid}, \tag{2.2}$$

where τ_1, and τ_2 are constant tuning factors. The relation 2.1 is somewhat similar to LBSNRank [JSW$^+$12] but the equation is specific to our approach.

Similarly, the distance sensitive rank of a location is defined as:

$$\Pi^{d}_{t_1,t_2}(l) = \alpha * \theta_{t_1,t_2}(l) + (1-\alpha) * \sum_{(l',l) \in E} \Pi^{d}_{t_1,t_2}(l'), \tag{2.3}$$

Terms	**Definition**
$\Pi^{\mathbf{a}}_{t_1,t_2}(l)$	rank of location l in the time range t_1, t_2 using the context $\mathbf{a}$
$\beta_{t_1,t_2}(l)$	categoric sensitive factor of location l in the time range t_1, t_2
$\theta_{t_1,t_2}(l)$	distance sensitive factor of location l in the time range t_1, t_2
$P(u, l, t_1, t_2)$	likelihood of checkin by user u to location l in the time range t_1, t_2
$V_{u,t_1,t_2}(l)$	visits by the user u to the location l, within the time interval t_1, t_2
$dist(l_1, l_2)$	distance between locations l_1 and l_2
U	the users in the dataset
L	the locations in the dataset
$l.cat$	category of the location l
ϵ	the threshold distance
α	the damping factor

Table 2.1: Terms used in the chapter for MAPS model

where $\theta_{t_1,t_2}(l)$ is the distance sensitive factor and is defined as:

$$\theta_{t_1,t_2}(l) = \gamma_1 * \frac{\sum\limits_{u \in U} \mid V_{u,t_1,t_2}(l) \mid}{\sum\limits_{u \in U, dist(l,l') \leq \epsilon} \mid V_{u,t_1,t_2}(l') \mid} + \gamma_2 * \frac{\sum\limits_{u \in U, dist(l,l') \leq \epsilon} \mid V_{u,t_1,t_2}(l') \mid}{\sum\limits_{p \in L, u \in U} \mid V_{u,t_1,t_2}(p) \mid}, \tag{2.4}$$

where γ_1 and γ_2 are constant tuning factors. The unified rank is the fusion of the two ranks and is defined as:

$$\Pi_{t_1,t_2}(l) = \xi_1 * \Pi^c_{t_1,t_2}(l) + \xi_2 * \Pi^d_{t_1,t_2}(l), \tag{2.5}$$

where ξ_1 and ξ_2 are tuning parameters for the two contexts.

$$\begin{aligned} P(u,l,t_1,t_2) = \Pi_{t_1,t_2}(l) * (\psi_d * & \sum_{\substack{l' \in L, \\ dist(l,l') \leq \epsilon}} \mid V_{u,t_1,t_2}(l') \mid \\ +\psi_c * & \sum_{\substack{l' \in L, \\ l.cat = l'.cat}} \mid V_{u,t_1,t_2}(l') \mid \\ +\psi_s * & \sum_{(u',u) \in friend} \mid V_{u',t_1,t_2}(l) \mid). \end{aligned} \tag{2.6}$$

The likelihood of check-in of user u at location l within a time t_1,t_2 is shown in Eqn. 2.6. The terms ψ_d, ψ_c, and ψ_s are defined using TF-IDF [SB88, WLWK08] for each user. For a user u,

$$\psi_d = \frac{n_d}{n} * log(1 + \frac{N}{N_d}), \tag{2.7}$$

where n_d is the number of visits by the user u that are within the threshold distance ϵ, n is the total visits count by u, N is the number of POIs, and N_d is the number of POIs that are within the threshold distance ϵ from the user's check-in history. For the categorial factor, we use the following relation:

$$\psi_c = \frac{n_c}{n}.log(1 + \frac{N}{N_c}), \tag{2.8}$$

where n_c is the number of visits by the user u to the category c, and N_c is the number of POIs with the category c. Similarly, for the social factor we define:

$$\psi_s = \frac{n_s}{n}.log(1 + \frac{N}{N_s}), \tag{2.9}$$

where n_s is the number of visits by the user u in common to her friends, and N_s is the number of visits in common to the friends for all the users $u \in U$.

According to the contexts used, we analyzed the performance of three different models, the categorical link based model (CLM) (defined in Eqn. 2.1 and Eqn. 2.2), the spatial link based model (SLM) (defined in Eqn. 2.3 and Eqn. 2.4), and the fused model MAPS (defined in Eqn. 2.5 and Eqn. 2.6).

2.4 Evaluation

In this section, we present the evaluation dataset, evaluation metrics, and the results and discussion on our findings.

2.4.1 DataSet

We used the Weeplaces and the Gowalla dataset [LLAM13] which was collected from the popular LBSNs Gowalla and Weeplaces. The statistics of the dataset are shown in Table 2.2.

Dataset	**Check-ins**	**Users**	**Locations**	**Links**	**Location Categories**
Gowalla	36,001,959	319,063	2,844,076	337,545	629
Weeplace	7,658,368	15,799	971,309	59,970	96

Table 2.2: Statistics of the dataset.

These datasets were well defined and had the attributes relevant to the context of the problem, such as, (i) location category, (ii) geospatial co-ordinates, (iii) friendship information, and (iv) check-in time. After avoiding incomplete records, the 5 most checked-in categories (and their check-in count) were: (i) Home/Work/Other: Corporate/Office (437,824), (ii) Food: Coffee Shop (267,589), (iii) Nightlife:Bar (248,565), (iv) Shop: Food & Drink: Grocery/Supermarket (161,016), and (v) Travel: Train Station (152,114) for Weeplaces, and (i) Corporate Office (1,750,707), (ii) Coffee Shop (1,063,961), (iii) Mall (958,285), (iv) Grocery (884,557), and (iv) Gas & Automotive (863,199) for the Gowalla dataset. The "work" or "home" related category (Home/Work/ Other: Corporate/Office) was popular from 6 am to 6 pm, with the highest check-ins (42,019) made at 1 pm. Similarly, the "bars" had highest of 21,806 check-ins at 2 am and the lowest check-ins (15,209) at 5 am. Most of the check-ins were at 12 pm - 6 pm and were done either in POIs that are of “Home” or “Work” related categories.

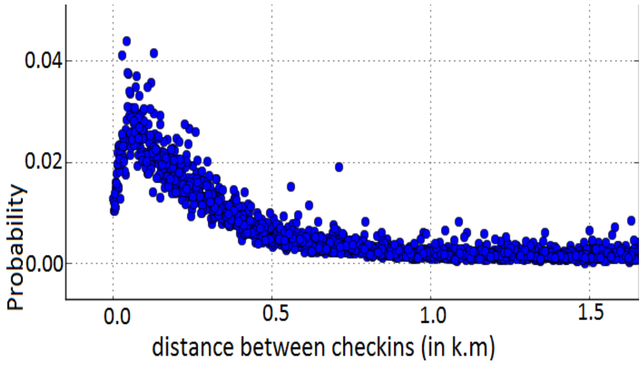


Figure 2.1: Impact of distance to check-in trend in Weeplaces dataset

Figure 2.1 illustrates the inverse relation of the distance to the check-in frequency. It was obtained by plotting the distance between the chronologically sorted consecutive check-ins of each user and the likelihood of the users' check-in in that distance (for ease, the distance was rounded to four decimals). The check-ins centralized within some distance (the dense patches within 0.5 km) illustrate the willingness to near places.

Models	**Precision**	**Recall**	**F-Score**
Ye et al. [YYLL11]	0.02417	0.00095	0.00183
LBSNRank [JSW$^+$12]	0.08496	0.00063	0.00125
Wang et al. [WTM13]	0.01818	0.00052	0.00101
CLM	0.00428	0.00024	0.00045
SLM	0.09085	0.00799	0.01468
MAPS	0.29769	0.01039	**0.02008***

Table 2.3: Average Performance of MAPS in Weeplaces dataset

Models	**Precision**	**Recall**	**F-Score**
Ye et al. [YYLL11]	0.03000	0.00120	0.00230
LBSNRank [JSW$^+$12]	0.40900	0.00300	0.00600
Wang et al. [WTM13]	0.10600	0.00200	0.00392
CLM	0.00633	0.00154	0.00247
SLM	0.25350	0.00973	0.01874
MAPS	0.35400	0.03100	**0.05700***

Table 2.4: Average Performance of MAPS in Gowalla dataset

2.4.2 Results and Discussions

A 5 -fold cross validation with top N (5, 10, 15 and 20) recommendation scores was used for the precision (P), the recall (R) and the F-score (2*P*R/(P+R)) metrics.

We used $\alpha = 0.85$ and the convergence was detected when the rank scores of the nodes were not changing anymore. For each model, the tuning parameters were selected from the trials conducted with three set of parameters ((0.25:0.75), (0.5:0.5), and (0.75:0.25)). The categoric model performed best when $\tau_1 = 0.75$ and $\tau_2 = 0.25$, and for distance model it was when $\gamma_1 = 0.75$ and $\gamma_2 = 0.25$. Similarly, among the three set of parameters the unified model performed best with the categorical context weight of 0.25.

Models	**Precision@N**	**Recall@N**
Ye et al. [YYLL11]	@5= 0.0303 @10= 0.0230 @15= 0.0191	@5= 0.0008 @10= 0.0009 @15= 0.0011
LBSNRank [JSW$^+$12]	@5= 0.0853 @10= 0.0848 @15= 0.4090	@5= 0.0006 @10= 0.0006 @15= 0.0030
Wang et al. [WTM13]	@5= 0.0449 @10= 0.0414 @15= 0.1060	@5= 0.0014 @10= 0.0020 @15= 0.0022
MAPS	@5= **0.2440** @10= **0.3050** @15= **0.3360**	@5= **0.0045** @10= **0.0092** @15= **0.0310**

Table 2.5: Precision@N, Recall@N of MAPS against other studies

The comparative performance of different models is illustrated in Table 2.3 and Table 2.4. The observed difference was statistically significant at 95% confidence level. Table 2.5 shows the average metrics across the top 5, 10, and 15 recommended items .

2.5 Analysis of roles of contexts

In the previous sections, we presented the analysis of check-in data based on (a) the categorical, (b) social, (c) spatial, and (d) temporal contexts, however we did not analyze the impact of the individual contexts. In this section, we define and analyze the fusion of different major contexts for POI recommendation. Such a fusion and analysis is barely explored by other researchers. The major contributions of this research are: (i) it analyzes the role of different contexts (e.g., check-in frequency, social, temporal, spatial, and categorical) in the location recommendation, (ii) it proposes two fused models -a ranking-based, and a matrix factorization-based, that incorporate all the major contexts into a single recommendation model, and (iii) it evaluates the proposed models against two real-world datasets.

Though some contexts might not be a sole contributor, combining them with other contexts might positively impact the recommendation quality. The role of above-mentioned contexts is illustrated in Figure 2.2. In the figure, two users u_1 and u_2 are friends. The *social* impact of user u_2 can influence user u_1 to the places that were visited by user u_2. The spatial influence can affect user u_1 to select the nearest location among the available options. As shown in the figure, the visit of user u_1 to a "cafe" around the same hour of days is due to temporal influence. Similarly, the categorical influence is reflected when a coffee lover visits any place that serves coffee. There can be additional influential factors, for instance, (i) the utility of a POI, regardless of the distance and cost, (ii) the popularity of a POI (due to historical, cultural, social or other impacts), (iii) the dynamic mobility of a user (trend to visit new places), (iv) promotional offers, such as coupons, discounts, and so forth. Though it might also be interesting to explore these factors, we defer them for our future studies.

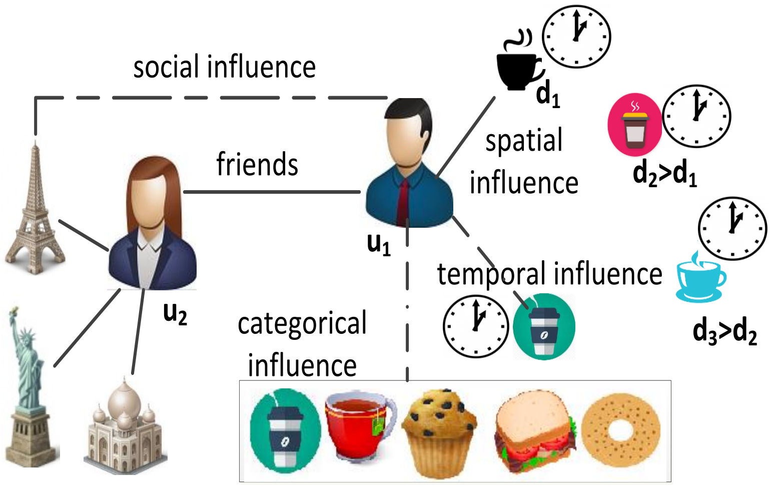


Figure 2.2: Illustration of different contextual influence in LBSN

We can see that most of the existing studies are focused on the check-in frequency and only few of them have exploited additional factors. Though the study from Stepan et al. [SMDM16] looks more relevant to our work, two of the major differences make our study more interesting. First, we incorporate the location category but they did not. Second, we analyze the roles of major contexts by combining different contexts in different fused models but their model analyzed the role of one context at a time and only fused the contexts in the final model. Our paper attempts to fill this gap via detailed analysis of impact of different factors (for instance, (i) what might be the impact of using social and temporal factors instead of spatial and temporal factors?, (ii) does the social factor contribute more than categorical factor?, (iii) can we get better results by having more factors?, and so on). To the best of our knowledge, none of the existing recommendation models spanned to incorporate all the major contexts. The exploitation of roles of different contexts and their incorporation into a single model is the novelty of our paper. We present ranking-based and matrix factorization-based fused models in this paper.

2.5.1 Ranking-based approach

In this section, we present ranking-based models that fuse the categorical, social, spatial, and temporal contexts.

1. Single context: This is based on check-in frequency (F). While it might be possible to get several single context models, we mainly focus on the check-in frequency because it is the basic criteria to be used for the prediction. The other contexts mainly act like a supplement to this context.

2. Two contexts: These models use check-in frequency along with some other contexts. The following models are defined in this paper: (i) check-in frequency and temporal (check-in time) (FT), (ii) check-in frequency and social (friends) (FS), (iii) check-in frequency and location category (FC), (iv) check-in frequency and spatial(location distance) (FD).

3. Three contexts: We combine three contexts to define following models: (i) the check-in frequency, social, and temporal (FST), (ii) the check-in frequency, categorical, and temporal (FCT), (iii) the check-in frequency, spatial, and temporal (FDT), (iv) the check-in frequency, categorical, and social (FCS), (v) the check-in frequency, spatial, and social (FDS), (vi) the check-in frequency, categorical, and spatial (FCD).

4. Four contexts: We define the following models: (i) check-in frequency, categorical, social, and temporal (FCST), (ii) check-in frequency, spatial, social, and temporal (FDST), (iii) check-in frequency, categorical, spatial, and temporal (FCDT), and (iv) check-in frequency, categorical, spatial, and social (FCDS).

5. Five contexts (FCDST): It fuses the check-in frequency, categorical, spatial, social, and temporal contexts.

All of the models proposed in this section are based on the Topic-Sensitive PageRank [Hav02] which can introduce some bias to the PageRank [PBMW99] vector. The Topic-Sensitive PageRank [Hav02] is a state-of-art ranking method for large graphs. It can incorporate the set of influential or representative (or additional context relevant attributes) topics to address the importance of particular topics. For a given context, it can identify the most closely associated (contextual) topics and such relevant topic-sensitive (biased) vectors can be used to rank the documents satisfying the query. It is a good fit for us because we can represent the users, locations, check-in relation, and social relation as a graph and use the additional factors of LBSN to achieve personalized ranking of user and place nodes. Similar to the web graph, we can assure the convergence of ranking of user-location graph by adding the damping factor $(1 - \alpha)$ to the rank propagation. This can improve quality of PageRank not only by limiting the effect of rank sinks [BMPW98], but also by assuring the convergence to a unique rank vector [Hav02].

We define 16 different recommendation models. The terms used in this section are defined in Table 2.6.

Single context

In this model, the check-in frequency of a location is solely used to define the popularity of a location. The rank of a location is then defined as:

$$\begin{aligned} \Pi^f(l) &= \alpha\beta^f(l) + (1-\alpha)\sum_{\substack{l' \in L \\ l' \neq l}} \Pi^f(l'), \\ \beta^f(l) &= \frac{\sum\limits_{u \in U} \mid V_u(l) \mid}{\sum\limits_{\substack{u \in U \\ p \in L}} \mid V_u(p) \mid}, \end{aligned} \tag{2.10}$$

where the term $\beta^f(l)$ is the check-in frequency personalization. The highly ranked locations can be recommended to the users. This approach will always recommend the same set of locations to all the users because the rank

Terms	**Definition**
U	the users in the dataset
L	the locations in the dataset
u_L	locations visited by the user u
u_f	the friends of the user u
l_U	visitors to the location l
$l.cat$	category of the location l
$V_u(l)$	visit counts of user u to the location l
$V_{u,t_1,t_2}(l)$	visit counts of user u to location l, in time range t_1, t_2
$dist(l_1, l_2)$	distance between locations l_1 and l_2
$\Pi^{\mathbf{a}}(l)$	rank of location l using the context **a**
$\Pi^{\mathbf{a}}_{t_1,t_2}(l)$	rank of location l in time range t_1, t_2 using the context **a**
$\beta_{t_1,t_2}(l)$	topic sensitive factor of location l in time range t_1, t_2
$\beta_{a,b}(l)$	topic sensitive factor of location l using the contexts a and b
$P(u, l, t_1, t_2)$	likelihood of checkin by user u to location l in time range t_1, t_2
K_i^a	constant parameters using context a
ψ_+^+	weight factor estimated by TF-IDF
α	the damping factor
ϵ	the threshold distance

Table 2.6: Definition of terms used in the chapter

of a location is only dependent on the frequency of check-ins across all the users. A better approach would be to personalize the recommendation by using similarity of the target location to the locations already visited by the user. The likelihood of a user u to visit a location l can be then defined as: $P^f_{u,l} = \Pi^f(l) * \psi^l_f$. The term ψ^l_f (defined later) is the weight factor which can be estimated using TF-IDF [SB88, WLWK08]. This model favors the locations with common visitors and assigns a non-zero, positive similarity value only to places with common visitors. So, in this case, the likelihood of visiting a location will depend only on its rank. Though the common visitors count might be used to measure the similarity between places, the places with common spatial, temporal, or categorical trend cannot be addressed with this model.

Two contexts

In this model, two contexts are incorporated to get the following fused models:

1. Categorical (FC): This approach ameliorates the recommendation model by incorporating the location category. The rank of a location is defined as:

$$\Pi^c(l) = \alpha\beta^c(l) + (1-\alpha)\sum_{\substack{l' \in L,\\ l' \neq l,\\ l.cat=l'.cat}} \Pi^c(l'),$$

$$\beta^c(l) = K_1^c * \frac{\sum\limits_{u \in U} \mid V_u(l) \mid}{\sum\limits_{\substack{l.cat=l'.cat\\ u \in U}} \mid V_u(l') \mid} + K_2^c * \frac{\sum\limits_{\substack{l.cat=l'.cat\\ u \in U}} \mid V_u(l') \mid}{\sum\limits_{\substack{u \in U\\ p \in L}} \mid V_u(p) \mid}, \tag{2.11}$$

where $\beta^c(l)$ is the categorical personalization, $K_1^c \in [0,1]$ and $K_2^c \in [0,1]$ are constants. The likelihood of a user (u) to visit a location (l) is then defined as: $P_{u,l}^c = \Pi^c(l) * \psi_c^l$, where ψ_c^l is estimated using TF-IDF.

2. Temporal (FT): Any two locations that have same check-in hour (or within a threshold time interval) can be more likely similar than the ones having check-in time beyond the threshold. The rank of a location can then be defined using the following relation:

$$\Pi^t(l) = \alpha\beta^t(l) + (1-\alpha)\sum_{\substack{l \in L,\\ l' \neq l,\\ l.t=l'.t}} \Pi^t(l'),$$

$$\beta^t(l) = K_1^t * \frac{\sum\limits_{u \in U} \mid V_{u,t}(l) \mid}{\sum\limits_{\substack{u \in U\\ l' \in L}} \mid V_{u,t}(l') \mid} + K_2^t * \frac{\sum\limits_{\substack{u \in U\\ l' \in L}} \mid V_{u,t}(l') \mid}{\sum\limits_{\substack{u \in U\\ p \in L}} \mid V_u(p) \mid}, \tag{2.12}$$

where $\beta^t(l)$ is the temporal personalization. The likelihood of check-in for user u, in location l, at time t is then defined as: $P^t_{u,l} = \Pi^t(l) * \psi^l_t$.

3. Spatial (FD): This model is influenced by Tobler's Law ("*everything is related to everything else, but near things are more related than the distant things*") [Tob70]. Using this and the distance context we define the rank as:

$$\Pi^d(l) = \alpha\beta^d(l) + (1-\alpha)\sum_{\substack{l'\in L,\\ l'\neq l,\\ dist(l,l')\leq\epsilon}} \Pi^d(l'),$$

$$\beta^d(l) = K^d_1 * \frac{\sum\limits_{u\in U} \mid V_u(l) \mid}{\sum\limits_{\substack{\text{dist(l,l')}\leq\epsilon\\ u\in U}} \mid V_u(l') \mid} + K^d_2 * \frac{\sum\limits_{\substack{\text{dist(l,l')}\leq\epsilon\\ u\in U}} \mid V_u(l') \mid}{\sum\limits_{\substack{u\in U\\ p\in L}} \mid V_u(p) \mid}, \tag{2.13}$$

where the term $\beta^d(l)$ is the spatial personalization, $K^d_1 \in [0,1]$ and $K^d_2 \in [0,1]$ are constants. The likelihood of a user u to visit a location l is then defined as: $P^d_{u,l} = \Pi^d(l) * \psi^l_d$.

4. Social (FS): Generally, the social networks define the social-tie between users (for instance, friends, followers, and so forth). Using this concept, we formulate the impact of social relation as:

$$\Pi^s(l) = \alpha\beta^s(l) + (1-\alpha)\sum_{\substack{u\in l_U,\\ u'\in u_f,\\ l'\in u'_L}} \Pi^s(l'),$$

$$\beta^s(l) = \sum_{u\in U}\left(K^s_1 * \frac{\sum\limits_{u'\in u_f} \mid V_{u'}(l) \mid}{\sum\limits_{\substack{u'\in u_f\\ l'\in L}} \mid V_{u'}(l') \mid} + K^s_2 * \frac{\sum\limits_{\substack{u'\in u_f\\ l'\in L}} \mid V_{u'}(l') \mid}{\sum\limits_{\substack{u\in U\\ p\in L}} \mid V_u(p) \mid}\right), \tag{2.14}$$

where $\beta^s(l)$ is the social personalization, $K^s_1 \in [0,1]$ and $K^s_2 \in [0,1]$ are constants. In this model, the popularity of a location is computed by taking into account the fraction of check-in counts it gets among the check-in counts in the friend circle. The likelihood of a user u to visit a location l is then defined as: $P^s_{u,l} = \Pi^s(l) * \psi^l_s$.

Three contexts

In this model, three contexts are incorporated to get the following fused models:

1. Categoric-Temporal (FCT): We define the categoric-temporal ranking as:

$$\Pi^{c}_{t_1,t_2}(l) = \alpha * \beta_{t_1,t_2}(l) + (1-\alpha) * \sum_{\substack{l' \in L, \\ l' \neq l, l'.cat = l.cat}} \Pi^{c}_{t_1,t_2}(l'),$$

$$\beta_{t_1,t_2}(l) = K^{ct}_1 * \frac{\sum\limits_{u \in U} \mid V_{u,t_1,t_2}(l) \mid}{\sum\limits_{\substack{l.cat = l'.cat \\ u \in U}} \mid V_{u,t_1,t_2}(l') \mid} + K^{ct}_2 * \frac{\sum\limits_{\substack{l.cat = l'.cat \\ u \in U}} \mid V_{u,t_1,t_2}(l') \mid}{\sum\limits_{p \in L, u \in U} \mid V_{u,t_1,t_2}(p) \mid}, \tag{2.15}$$

where $\beta_{t_1,t_2}(l)$ is the categoric sensitive factor. The likelihood of a user (u) to visit a location (l) is then defined as: $P^{ct}_{u,l} = \Pi^{c}_{t_1,t_2}(l) * \psi^{l}_{ct}$.

2. Socio-Temporal (FST): This model is defined by substituting the categorical constraint with social constraint in the Categoric-Temporal (FCT) model.

3. Spatio-Temporal (FDT): This model is defined by substituting categorical context with spatial in the Categoric-Temporal (FCT) model.

4. Categorical-Spatial (FCD): This model uses the categorical and spatial factors to rank the locations as:

$$\Pi^{d}_{c}(l) = \alpha * \beta_{cd}(l) + (1-\alpha) * \sum_{\substack{l' \in L, \\ l' \neq l, \\ dist(l',l) \leq \epsilon, \\ l'.cat = l.cat}} \Pi^{d}_{c}(l'),$$

$$\beta_{cd}(l) = K^{cd}_1 * \frac{\sum\limits_{u \in U} \mid V_u(l) \mid}{\sum\limits_{\substack{u \in U \\ dist(l,l') \leq \epsilon \\ l.cat = l'.cat}} \mid V_u(l') \mid} + K^{cd}_2 * \frac{\sum\limits_{\substack{u \in U \\ dist(l,l') \leq \epsilon \\ l.cat = l'.cat}} \mid V_u(l') \mid}{\sum\limits_{p \in L, u \in U} \mid V_u(p) \mid}, \tag{2.16}$$

where $\beta_{cd}(l)$ is the categoric-distance sensitive factor, K^{cd}_1, and K^{cd}_2 are constant tuning factors. The likelihood of a user to visit a location is then defined as: $P^{cd}_{u,l} = \Pi^{d}_{c}(l) * \psi^{l}_{cd}$.

5. Spatio-Social (FDS): The ranking of a location in terms of spatial and social factors is defined as:

$$\Pi_s^d(l) = \alpha * \beta_{sd}(l) + (1-\alpha) * \sum_{\substack{l' \in L, \\ dist(l',l) \leq \epsilon, \\ u' \in l_U, \\ l' \in u'_{f_L}}} \Pi_s^d(l'),$$

$$\beta_{sd}(l) = \sum_{u \in U} \Big(K_1^{sd} * \frac{\sum\limits_{u' \in u_f} \mid V_{u'}(l) \mid}{\sum\limits_{\substack{u' \in u_f \\ dist(l,l') \leq \epsilon}} \mid V_{u'}(l') \mid} + K_2^{sd} * \frac{\sum\limits_{\substack{dist(l,l') \leq \epsilon \\ u' \in u_f}} \mid V_{u'}(l') \mid}{\sum\limits_{\substack{p \in L \\ u' \in u_f}} \mid V_u(p) \mid} \Big), \tag{2.17}$$

where $\beta_{sd}(l)$ is the spatio-social sensitive factor, K_1^{sd}, and K_2^{sd} are constant tuning factors. The likelihood of a user to visit a location is then defined as: $P_{u,l}^{sd} = \Pi_s^d(l) * \psi_{ds}^l$.

6. Categorical-Social (FCS): This model can be defined by substituting the spatial constraint with categorical constraint in the definition of Spatio-Social (FDS) model.

Four contexts

These recommendation models have three other contexts along with the check-in frequency.

1. Categoric-Spatial-Temporal (FCDT): In this model, the categorical, spatial, and temporal contexts are incorporated into a single recommendation model. The rank of a location can then be defined as:

$$\Pi_{t_1,t_2}^{cd}(l) = \alpha * \beta_{cdt}(l) + (1-\alpha) * \sum_{\substack{dist(l',l) \leq \epsilon, \\ l'.cat=l.cat}} \Pi_{t_1,t_2}^{cs}(l'),$$

$$\beta_{cdt}(l) = K_{cdt}^1 * \frac{\sum\limits_{u \in U} \mid V_{u,t_1,t_2}(l) \mid}{\sum\limits_{\substack{l.cat=l'.cat \\ dist(l,l') \leq \epsilon \\ u \in U}} \mid V_{u,t_1,t_2}(l') \mid} + K_{cdt}^2 * \frac{\sum\limits_{\substack{l.cat=l'.cat \\ dist(l,l') \leq \epsilon \\ u \in U}} \mid V_{u,t_1,t_2}(l') \mid}{\sum\limits_{p \in L, u \in U} \mid V_{u,t_1,t_2}(p) \mid}, \tag{2.18}$$

where $\beta_{cdt}(l)$ is the categoric-spatial-temporal sensitive factor, K^1_{cdt} and K^2_{cdt} are constant tuning factors. The likelihood of a user to visit a location is then defined as: $P^{cdt}_{u,l} = \Pi^{cd}_{t_1,t_2}(l) * \psi^l_{cdt}$.

2. Categoric-Spatial-Social (FCDS): In this model, the categorical, spatial, and social contexts are incorporated into a single recommendation model. The rank of a location can then be defined as:

$$\Pi^{cs}_d(l) = \alpha * \beta_{cds}(l) + (1-\alpha) * \sum_{\substack{dist(l',l)\leq\epsilon,\\ l'.cat=l.cat,\\ u\in l_U,\\ l'\in u_{f_L}}} \Pi^{cs}_d(l'),$$

$$\beta_{cds}(l) = \sum_{u\in U} \Bigg(K^{cds}_1 * \frac{\sum\limits_{u'\in u_f} \mid V_{u'}(l) \mid}{\sum\limits_{\substack{l.cat=l'.cat\\ dist(l,l')\leq\epsilon\\ u'\in u_f\\ l'\in L}} \mid V_{u'}(l') \mid} + K^{cds}_2 * \frac{\sum\limits_{\substack{l.cat=l'.cat\\ dist(l,l')\leq\epsilon\\ u'\in u_f\\ l'\in L}} \mid V_{u'}(l') \mid}{\sum\limits_{\substack{p\in L\\ u\in U}} \mid V_u(p) \mid} \Bigg), \tag{2.19}$$

where $\beta_{cds}(l)$ is the categoric-spatial-social sensitive factor, K^{cds}_1 and K^{cds}_2 are constant tuning factors.

3. Categoric-Social-Temporal (FCST): In this model, the categorical, social, and temporal contexts are incorporated into a single recommendation model. The rank of a location can then be defined as:

$$\Pi^{cs}_{t_1,t_2}(l) = \alpha * \beta_{cst}(l) + (1-\alpha) * \sum_{\substack{l'.cat=l.cat,\\ u\in l_U,\\ l'\in u_{f_L},\\ l'\in L}} \Pi^{cs}_{t_1,t_2}(l'),$$

$$\beta_{cst}(l) = \sum_{u\in U} (K^{cst}_1 * \frac{\sum\limits_{u'\in u_f} \mid V_{u',t_1,t_2}(l) \mid}{\sum\limits_{\substack{l.cat=l'.cat\\ u'\in u_f}} \mid V_{u',t_1,t_2}(l') \mid} + K^{cst}_2 * \frac{\sum\limits_{\substack{l.cat=l'.cat\\ u'\in u_f}} \mid V_{u',t_1,t_2}(l') \mid}{\sum\limits_{\substack{p\in L\\ u\in U}} \mid V_{u,t_1,t_2}(p) \mid}), \tag{2.20}$$

where $\beta_{cst}(l)$ is the categoric-social-temporal sensitive factor, K^{cst}_1 and K^{cst}_2 are constant tuning factors, and $t_1 \leq t \leq t_2$. The likelihood of a user to visit a location is then defined as: $P^{cst}_{u,l} = \Pi^{cs}_{t_1,t_2}(l) * \psi^l_{cst}$.

4. Spatial-Social-Temporal (FDST): This model can be defined by substituting the categorical constraint with spatial constraint in the definition of Categoric-Social-Temporal (FCST) model.

Five contexts

In this model (FCDST), all the major contexts (e.g., categorical, social, spatial, and temporal) along with the check-in frequency are incorporated into a single model. The categorical sensitive ranking is defined as:

$$\Pi^c_{t_1,t_2}(l) = \alpha * \beta_{t_1,t_2}(l) + (1-\alpha) * \sum_{\substack{l'.cat=l.cat,\\ l' \in L,\\ l' \neq l}} \Pi^c_{t_1,t_2}(l'),$$

$$\beta_{t_1,t_2}(l) = K^c_1 * \frac{\sum\limits_{u \in U} \mid V_{u,t_1,t_2}(l) \mid}{\sum\limits_{\substack{l.cat=l'.cat\\ u \in U}} \mid V_{u,t_1,t_2}(l') \mid} + K^c_2 * \frac{\sum\limits_{\substack{l.cat=l'.cat\\ u \in U}} \mid V_{u,t_1,t_2}(l') \mid}{\sum\limits_{\substack{p \in L\\ u \in U}} \mid V_{u,t_1,t_2}(p) \mid}, \tag{2.21}$$

where $\beta_{t_1,t_2}(l)$ is the categoric sensitive factor, K^c_1, and K^c_2 are constant tuning factors.

Similarly, the distance sensitive rank of a location is defined as:

$$\Pi^d_{t_1,t_2}(l) = \alpha * \theta_{t_1,t_2}(l) + (1-\alpha) * \sum_{\substack{dist(l',l) \leq \epsilon\\ l' \in L\\ l' \neq l}} \Pi^d_{t_1,t_2}(l'),$$

$$\theta_{t_1,t_2}(l) = K^d_1 * \frac{\sum\limits_{u \in U} \mid V_{u,t_1,t_2}(l) \mid}{\sum\limits_{\substack{dist(l,l') \leq \epsilon\\ u \in U}} \mid V_{u,t_1,t_2}(l') \mid} + K^d_2 * \frac{\sum\limits_{\substack{dist(l,l') \leq \epsilon\\ u \in U}} \mid V_{u,t_1,t_2}(l') \mid}{\sum\limits_{\substack{p \in L\\ u \in U}} \mid V_{u,t_1,t_2}(p) \mid}, \tag{2.22}$$

where $\theta_{t_1,t_2}(l)$ is the distance sensitive factor, K^d_1, and K^d_2 are constant tuning factors. The unified rank is the fusion of the two ranks and is defined as:

$$\Pi_{t_1,t_2}(l) = \xi_1 * \Pi^c_{t_1,t_2}(l) + \xi_2 * \Pi^d_{t_1,t_2}(l), \tag{2.23}$$

where ξ_1, ξ_2 are tuning parameters for the two contexts. The likelihood of check-in for the user u at location l within the time frame t_1,t_2 is defined as:

$$P(u,l,t_1,t_2) = \Pi_{t_1,t_2}(l) * \Big(\psi_d * \sum_{\substack{l' \in L,\\ dist(l,l') \le \epsilon}} \mid V_{u,t_1,t_2}(l') \mid + \psi_c * \sum_{\substack{l' \in L,\\ l.cat = l'.cat}} \mid V_{u,t_1,t_2}(l') \mid \\ + \psi_s * \sum_{(u',u) \in friend} \mid V_{u',t_1,t_2}(l) \mid \Big), \tag{2.24}$$

where the terms ψ_* are estimated using TF-IDF.

Parameters Estimation

The parameters used in the likelihood relations are defined using TF-IDF [SB88, WLWK08] for each user. For a user u, $\psi_d = \frac{n_d}{n}.log(1 + \frac{N}{N_d})$, where n_d is the number of visits by user u that are within the threshold distance ϵ, and N_d is the number of POIs that are within the threshold distance ϵ from the user's check-in history, n is the total visits made by user u, and N is the number of POIs. For the categorical factor, we use the relation: $\psi_c = \frac{n_c}{n}.log(1 + \frac{N}{N_c})$, where n_c is the number of visits by user u to category c, and N_c is the number of POIs with the category c. Similarly, $\psi_s = \frac{n_s}{n}.log(1 + \frac{N}{N_s})$, where n_s is the number of visits by a user u in common to her friends, and N_s is the number of visits in common to the friends for all the users $u \in U$. The other parameters are defined accordingly.

2.5.2 Matrix Factorization-based approach

After demonstrating its effectiveness in the Netflix Prize competition[4], the Matrix Factorization [KBV09] technique has been widely renowned in recommendation domain. The basic factorization model attempts to predict the user-item rating by mapping the original rating matrix into low dimensional

[4] http://www.netflixprize.com/

latent factor matrices. Given r_{ui} as the rating of a user u to the item i, the basic idea is to approximate the rating by using the lower order latent user matrix $\mathbf{p}$ and latent item matrix $\mathbf{q}$ which can be realized as: $\hat{r}_{ui} = q_i^T.p_u$. Basically, the entries at q_i represent the extent to which the item i possesses these factors, and the entries at p_u represent the extent of preference of user u on the items that are high on these factors. The main intuition is to minimize the following objective function by regularizing the above relation as:

$$\min_{q,p} \sum_{(u,i)\in k} (r_{ui} - q_i^T p_u)^2 + \lambda(|| \ q_i \ ||^2 + || \ p_u \ ||^2). \tag{2.25}$$

where k is the number of user-item pairs whose rating is known in the training set, and λ is a regularization constant.

For the POI domain, the check-in frequency can be taken as an implicit rating. Inspired from [KBV09], we extend the above relation to incorporate additional factors as mentioned below:

$$\hat{r}_{ui}(t) = \mu + b_i(t) + b_u(t) + q_i^T[p_u(t) + \sum_{a\in A(u)} y_a + \sum_{l\in u_L} x_l], \tag{2.26}$$

where $\hat{r}_{ui}(t)$ is the estimated rating of a user u to the item i at time t, μ is the global average rating of all places, $b_i(t)$ is the location bias at time t (the difference of rating of location i to the average rating μ of all locations for the visits made at time t), $b_u(t)$ is the user bias, $A(u)$ is the set of user attributes, and x_l represent the factors of locations visited by the user.

For user attributes, we use the vector $< r_{cat_1}, r_{cat_2}, ..., r_{cat_k}, r_{soc}, r_{dist} >$, where for a user u, $r_{cat_1} = \frac{\sum_{l.cat=cat_1} V_u(l)}{\sum_{l'\in u_L} V_u(l)}$ is the ratio of total check-ins made to the places with category cat_1 to that of total check-ins made on all places, $r_{soc} = \frac{\sum_{l\in u_{f_L}} V_u(l)}{\sum_{l'\in u_L} V_u(l')}$ is the ratio of total check-ins made on the places visited due to social influence to that of total check-ins on all places, and $r_{dist} = \frac{\sum_{dist(l)\leq\epsilon} V_u(l)}{\sum_{l'\in u_L} V_u(l')}$ is the ratio of total check-ins on the places within a threshold distance ϵ (from users home or work place) to that of total check-ins on all places. Similarly, we use the vector $< r_{cat}, r_{soc}, r_{dist} >$ for places, where r_{cat} is the ratio of number of check-ins made to this place to that of check-ins made to places with the same category, r_{dist} is the ratio of number of check-ins made to this place to that of check-ins made in its nearby places, and r_{soc} is the fraction of check-ins due to social influence of the visitors of this place. Furthermore, these attributes can be time constrained by accounting only the check-ins within a time interval.

2.5.3 Evaluation

This section defines the dataset, analysis of the contexts, and the performance of different models. We evaluated the ranking-based models as defined earlier, Non-negative Matrix Factorization (simple) that just used the check-in frequency, Non-negative Matrix Factorization (fused) that incorporated additional factors (see Eqn. 2.26, and three relevant models [YYLL11], [JSW$^+$12], and [WTM13]). For Matrix Factorization, the check-in count of every user to a place was normalized in the range (0,10), the non-negative singular value decomposition [BG08] was used for initialization, and up to 5,000 iterations were observed. The hour of a day was used to analyze the temporal trend.

Dataset

The Weeplaces and the Gowalla datasets were collected from the popular LBSNs - Gowalla and the Weeplaces [LLAM13]. We found that the datasets were well-defined and also had all the attributes (the location category, geo-spatial coordinates, friendship information, and the check-in time) relevant to our model. The incomplete records were eliminated in the analysis and evaluation. The Gowalla dataset had records from November 2010 to June 1, 2011, and had only 7 main location categories, so we used the well defined subcategories instead. The statistics of the datasets is defined in Table 2.2.

Impact of distance to the check-ins

For every user, the check-ins were chronologically sorted and the distance between consecutive check-ins of each user was computed. The likelihood of a user to check-in at a particular distance (for convenience, the distance was arbitrarily rounded to four decimals) was estimated by her visit history. The inverse relation of check-in trend to the distance of POI in Weeplaces dataset (though the trend on Gowalla dataset is not shown, it also had similar trend) is illustrated in Figure 2.1. We can see that most of the users' check-ins are centralized within some distance (the dense patches within 0.5 k.m indicate that most of the users' had the check-ins to the near places). The figure also shows that the willingness of check-in decreases with the increasing distance of the location.

Distribution of check-ins based on location category and check-in time

The top-10 visited location categories (and their check-in counts) for Weeplace were: (i) Home/Work /Other: Corporate/Office (437,730), (ii) Home/Work /Other: Home (306,105), (iii) Food:Coffee Shop (267,572), (iv) Nightlife:Bar (248,563), (v) Shops: Food & Drink:Grocery Supermarket (160,913), (vi) Travel:Train Station (152,104), (vii) Food:Cafe (129,205), (viii) Travel:Subway (107,879), (ix) Food: American (100,174), and (x) Travel:Airport (98,183). Similarly, for Gowalla, we had: (i) Corporate Office(1,660,159), (ii) Coffee Shop (988,999), (iii) Mall (872,873), (iv) Grocery (820,326), (v) Gas&Automotive (806,916), (vi) Apartment (753,547), (vii) Asian (735,453), (viii) Train Station (680,612), (ix) Other - Food (665,229), and (x) American (634,031).

The *work* or *home* related category (Home / Work / Other: Corporate / Office) was popular from 6 am to 6 pm, with the highest check-ins (42,019) made at 1 pm. Similarly, the “bars” had highest of 21,806 check-ins at 2 am and the lowest check-ins (15,209) at 5 am. Most of the check-ins were made between 12 pm - 6 pm and were in either "Home" or "Work" related categories. Figure 2.3 and Figure 2.4 illustrate the hourly distribution of top-5 categories. The location categories are abbreviated as: HWOC = Home /Work /Other:Corporate /Office, HWO = Home/Work/Other : Home, FCS=Food: coffee: Shop, NB = Nightlife:Bar, SFG = Shops:Food & Drink:Grocery/Supermarket, CO = Corporate Office, CS = Coffee Shop, M = Mall, G = Grocery, G&A = Gas & Automotive.

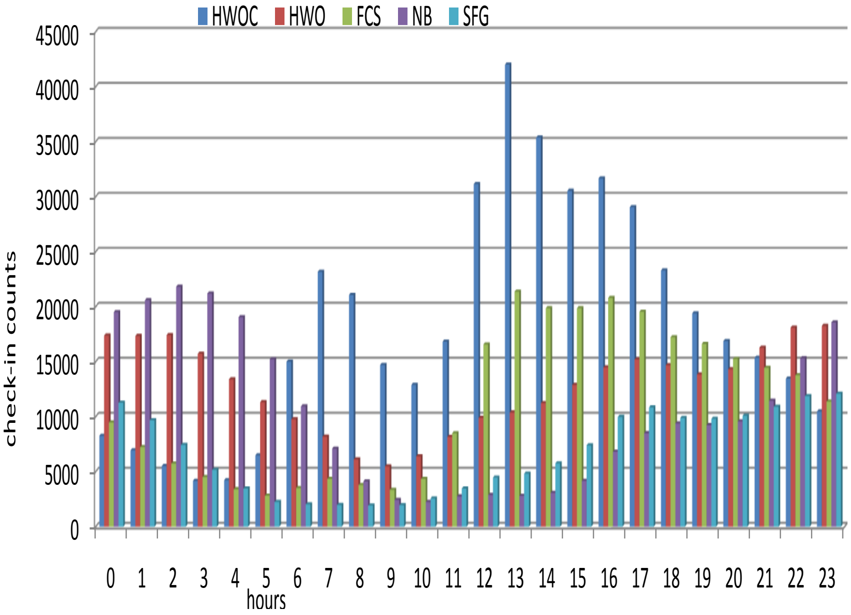


Figure 2.3: Hourly check-ins distribution for top - 5 categories in Weeplace dataset

Distribution of check-ins based on place

The top 10 places (and check-in counts) for Weeplace dataset were: (1) jr (13,769), (2) seoul (10,973), (3) san-francisco-international- airport-sfo-san-franci (10,658), (4) starbucks -new-york (10,329), (5) new-york-penn-station-new-york (7,809), (6) los-angeles-international- airport-lax-los-angeles (5,859), (7) grand-central-terminal-new-york (5,668), (8) john-f-kennedy-international-airport-jfk-queens (5,360), (9) whole-foods-new-york (4,562), and (10) station-

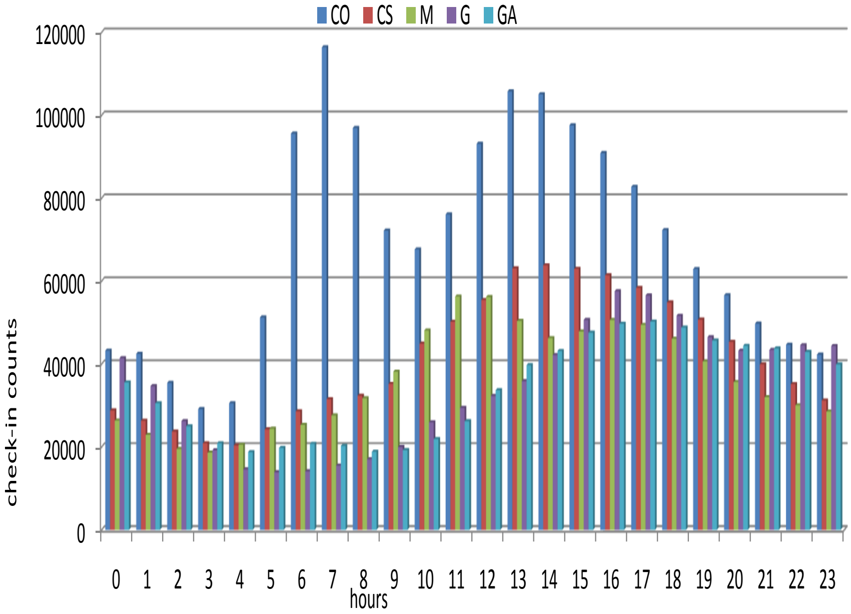


Figure 2.4: Hourly check-ins distribution for top - 5 categories in Gowalla dataset

utrecht-centraal-utrecht (4,227). Similarly the top 10 places (and check-in counts) for Gowalla[5] dataset were: (1) 55033 (28,414), (2) 19542 (19,996), (3) 66171 (19,186), (4) 9410 (18,542), (5) 58725 (18,457), (6) 23519 (18,136), (7) 10259 (17,397), (8) 9246 (15,909), (9) 155746 (15,640), and (10) 10190 (14,127).

Distribution of check-ins based on user

The check-in count of top-10 users in Weeplace dataset were: 1) thadd-fiala (6,517), (2) boon-yap (5,573), (3) eric-andersen (5,394), (4) sandro-pigoni (5,055), (5) john-lyons (4,963), (6) bob-boles (4,560), (7) hillary-lannan (4,342), (8) nate-folkert (4,289), (9) jason-allen (4,279), and (10) rue (4,237). Similarly, for the Gowalla dataset, the top-10 users were: 1) 84414 (45,375), (2) 213489 (44,960), (3) 269889 (44,726), (4) 27125 (41,017), (5) 30603 (33,851), (6)9298 (32,791), (7) 114774 (32,347), (8) 5153 (29,075), (9) 76390 (28,636), and (10)

[5] The Gowalla place id is numeric.

28509 (28,194). The hourly check-in distribution of top-5 visitors is shown in Figure 2.5 and Figure 2.6[6].

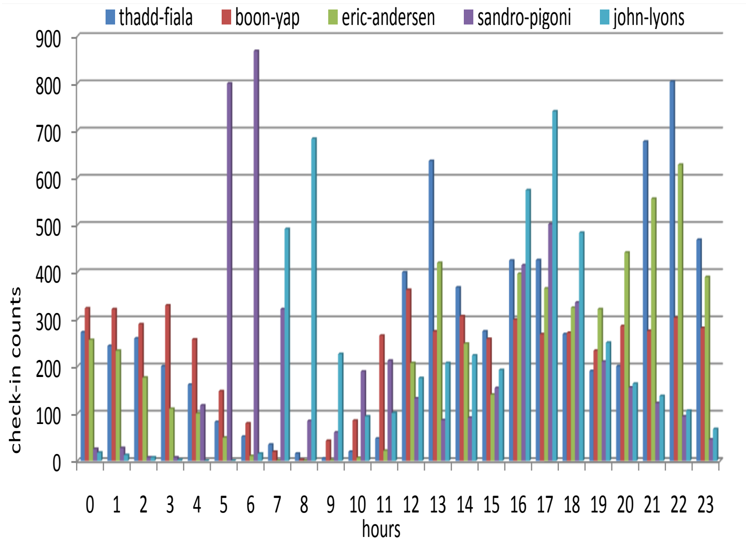


Figure 2.5: Hourly check-in distribution of top - 5 users in Weeplace dataset

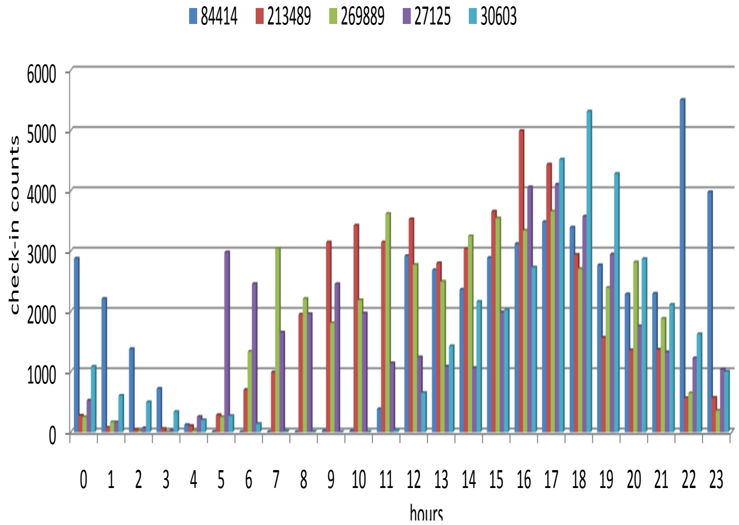


Figure 2.6: Hourly check-in distribution of top - 5 users in Gowalla dataset

Distribution of check-ins based on hour of a day

The hourly distribution of check-in counts are shown in Table 2.7. We present the hours of a day as in the range 0 to 23.

[6] The Gowalla user id is numeric.

Hours	**00**	**01**	**02**	**03**	**04**	**05**	**06**	**07**
D_1	396,066	353,423	316,200	274,667	224,824	199,217	206,555	210,419
D_2	1593,460	1,404,780	1,178,790	981,032	870,534	896,153	932,948	941,970
Hours	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**
D_1	185,940	170,621	196,994	240,939	305,864	332,520	328,622	362,591
D_2	887,360	925,178	1,116,408	1,294,830	1,445,426	1,569,247	1,691,046	1,886,372
Hours	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**
D_1	448,212	469,033	428,219	404,414	380,106	378,907	416,496	430,522
D_2	2,143,873	2,260,887	2,153,428	1,971,677	1,808,640	1,691,609	1,677,072	1,682,730

Table 2.7: Hourly check-in distribution for D_1 = Weeplaces and D_2 = Gowalla dataset.

Experimental Results

We used a 5 - fold cross validation and considered top N (5, 10, 15 and 20) recommendation scores to compute the precision ($P = \frac{|\text{true positive}|}{|\text{true positive}|+|\text{false positive}|}$), recall ($R = \frac{|\text{true positive}|}{|\text{true positive}|+|\text{false negative}|}$), and F-score (2*P*R/(P+R)) metrics. Though the goal of this paper is just to exploit the role of different contexts and not to compare the ranking based models with matrix factorization based models, we still illustrate the performance of these models in this section. The average performance of different models is illustrated in Table 2.8 and Table 2.9. The average metrics across the top@N recommendations are illustrated in Table 2.10 and Table 2.11.

Experimental setup

We used Python 2.7[7], Pandas 0.19.1[8], and Networkx 2.0[9] in a 24 core 2.40 GHz Intel(R) Xeon(R) CPU E5-2430L v2 CPU, 32 GB RAM, and a Scientific Linux release 6.5 (Carbon) for development and evaluation.

[7] https://www.python.org

[8] http://www.pandas.pydata.org

[9] https://www.networkx.github.io

Models	**Precision**	**Recall**	**F-Score**
F	0.067110	0.001175	0.002308
FC	0.003000	0.002140	0.002498
FS	0.064100	0.001900	*0.003690*
FT	0.026957	0.001804	0.003381
FD	0.035578	0.001100	0.002134
FCT	0.050933	0.000993	0.001948
FDT	0.091116	0.008065	*0.014818*
FCD	0.046334	0.001258	0.002449
FCS	0.079333	0.001683	0.003297
FDS	0.078166	0.002966	0.005716
FST	0.066260	0.007533	0.013528
FCST	0.094066	0.007700	0.014234
FDST	0.090636	0.006667	0.012420
FCDT	0.094166	0.008556	0.015687
FCDS	0.094166	0.008566	*0.015704*
FCDST	0.297690	0.010390	*0.020080*
Ye et al. [YYLL11]	0.024170	0.000950	0.001834
Jin et al. [JSW$^+$12]	0.084969	0.000639	0.001268
Wang et al. [WTM13]	0.018180	0.000520	0.001010
Simple Matrix Factorization	0.012747	0.044715	0.019838
Fused Matrix Factorization	0.330390	0.021793	**0.040888***

Table 2.8: Average performance of fused models in Weeplaces dataset

Parameter Analysis

We observed that the greater value of α resulted in slow convergence, had more impact from the inbound links, and had more evenly distributed impact to the nodes on outgoing links. This means the places with more visitors could be impacted with the higher value of α. We used $\alpha = 0.85$ which is a standard value for most graphs. The convergence was detected when the rank scores of the nodes were not changing anymore. The distance threshold ϵ was set to 1 k.m., which was simply based on the observation of the spatial check-in pattern of the users. For each model, the tuning parameters were selected from random trials conducted with three set of parameters ((0.25:0.75), (0.5:0.5), and (0.75:0.25)). The categorical module performed best when K_1^c

Models	**Precision**	**Recall**	**F-Score**
F	0.099857	0.001523	0.003000
FC	0.006370	0.001970	0.003009
FS	0.100708	0.004261	*0.008176*
FT	0.032011	0.002529	0.004687
FD	0.047063	0.001446	0.002806
FCT	0.061397	0.001591	0.003102
FDT	0.102230	0.009005	*0.016552*
FCD	0.048900	0.002438	0.004644
FCS	0.083138	0.003764	0.007202
FDS	0.078144	0.003884	0.007401
FST	0.078210	0.007611	0.013872
FCST	0.107462	0.008802	0.016271
FDST	0.124064	0.010270	0.018970
FCDT	0.129663	0.021236	*0.036495*
FCDS	0.106753	0.009580	0.017582
FCDST	0.354013	0.031066	*0.057120*
Ye et al. [YYLL11]	0.030000	0.001200	0.002307
Jin et al. [JSW$^+$12]	0.409000	0.003000	0.005956
Wang et al. [WTM13]	0.106000	0.002000	0.003925
Simple Matrix Factorization	0.014614	0.042218	0.021712
Fused Matrix Factorization	0.392227	0.038130	**0.069503***

Table 2.9: Average performance of fused models in Gowalla dataset

was 0.75. This implies the higher importance of the categorical popularity of a place than that of the popularity of the category. In other words, though the categorical factor can be influential, places with same category might have different popularity. For instance, one coffee shop might dominate the coffee business of a community. The spatial module performed best when K_1^d was 0.75. This implies the popularity of a location in its locality is of higher importance than the popularity of the whole locality itself. In other words, all places within a community may not have similar popularity. The five context model performed best when ξ_1 was 0.25. The weights for other modules were selected accordingly. The observed difference was statistically significant at 95% confidence level.

F	**FC**	**FS**	**FT**	**FD**
Precision@N @5= 0.100573 @10= 0.055769 @15= 0.044989 *Recall@N* @5= 0.002856 @10= 0.000304 @15= 0.000364	*Precision@N* @5= 0.002380 @10= 0.002441 @15= 0.004200 *Recall@N* @5= 0.001790 @10= 0.001635 @15= 0.003000	*Precision@N* @5= 0.044900 @10= 0.041400 @15= 0.106000 *Recall@N* @5= 0.001400 @10= 0.002000 @15= 0.002300	*Precision@N* @5= 0.029053 @10= 0.029693 @15= 0.022126 *Recall@N* @5= 0.001052 @10= 0.002170 @15= 0.002190	*Precision@N* @5= 0.025729 @10= 0.030010 @15= 0.051000 *Recall@N* @5= 0.000890 @10= 0.000982 @15= 0.001340
FCT	**FDT**	**FCD**	**FCS**	**FDS**
Precision@N @5= 0.042800 @10= 0.045000 @15= 0.065000 *Recall@N* @5= 0.000840 @10= 0.000860 @15= 0.001279	*Precision@N* @5= 0.090850 @10= 0.091100 @15= 0.091400 *Recall@N* @5= 0.007990 @10= 0.008100 @15= 0.008106	*Precision@N* @5= 0.037000 @10= 0.045000 @15= 0.057000 *Recall@N* @5= 0.000824 @10= 0.000950 @15= 0.002000	*Precision@N* @5= 0.077000 @10= 0.079000 @15= 0.082000 *Recall@N* @5= 0.001700 @10= 0.001100 @15= 0.002251	*Precision@N* @5= 0.067500 @10= 0.079000 @15= 0.088000 *Recall@N* @5= 0.001500 @10= 0.003500 @15= 0.003900
FST	**FCST**	**FDST**	**FCDT**	**FCDS**
Precision@N @5= 0.052100 @10= 0.067790 @15= 0.078880 *Recall@N* @5= 0.005910 @10= 0.007990 @15= 0.008700	*Precision@N* @5= 0.091200 @10= 0.094000 @15= 0.097000 *Recall@N* @5= 0.007600 @10= 0.007600 @15= 0.007900	*Precision@N* @5= 0.090330 @10= 0.090700 @15= 0.090880 *Recall@N* @5= 0.006560 @10= 0.006660 @15= 0.006760	*Precision@N* @5= 0.091100 @10= 0.093500 @15= 0.097900 *Recall@N* @5= 0.008210 @10= 0.008490 @15= 0.008970	*Precision@N* @5= 0.091100 @10= 0.093500 @15= 0.097900 *Recall@N* @5= 0.008227 @10= 0.008493 @15= 0.008980
FCDST	**[YYLL11]**	**[JSW$^+$12]**	**[WTM13]**	**Fused Matrix Factorization**
Precision@N @5= 0.244000 @10= 0.305000 @15= 0.336000 *Recall@N* @5= 0.004500 @10= 0.009200 @15= 0.031000	*Precision@N* @5= 0.030300 @10= 0.023020 @15= 0.019180 *Recall@N* @5= 0.000800 @10= 0.000900 @15= 0.001160	*Precision@N* @5= 0.085300 @10= 0.084800 @15= 0.409000 *Recall@N* @5= 0.000610 @10= 0.000610 @15= 0.003000	*Precision@N* @5= 0.0449 @10= 0.0414 @15= 0.1060 *Recall@N* @5= 0.0014 @10= 0.0020 @15= 0.0022	*Precision@N* @5= 0.291260 @10= 0.31995 @15= 0.375890 *Recall@N* @5= 0.019793 @10= 0.021992 @15= 0.022651

Table 2.10: Precision@N, Recall@N of different models in Weeplace dataset

Results and Discussion

1. We compared the performance of different models using the Precision, Recall, and F-Score metrics. From the evaluation (see Table 2.8 - 2.9), we observed that the fused models performed better than just the simple check-in frequency-based model. We can see that the quality of recommendation not only relied on the number of contexts fused, but also on the importance of the contexts fused.

F	**FC**	**FS**	**FT**	**FD**
Precision@N @5= 0.176573 @10= 0.057000 @15= 0.066000 *Recall@N* @5= 0.003721 @10= 0.000462 @15= 0.000387	*Precision@N* @5= 0.004897 @10= 0.005972 @15= 0.008231 *Recall@N* @5= 0.000993 @10= 0.001940 @15= 0.002970	*Precision@N* @5= 0.044595 @10= 0.05771 @15= 0.19982 *Recall@N* @5= 0.003542 @10= 0.003820 @15= 0.005422	*Precision@N* @5= 0.029820 @10= 0.031877 @15= 0.034337 *Recall@N* @5= 0.001577 @10= 0.002878 @15= 0.003133	*Precision@N* @5= 0.037864 @10= 0.038882 @15= 0.064443 *Recall@N* @5= 0.000721 @10= 0.000858 @15= 0.00276
FCT	**FDT**	**FCD**	**FCS**	**FDS**
Precision@N @5= 0.041983 @10= 0.05121 @15= 0.091000 *Recall@N* @5= 0.000899 @10= 0.001776 @15= 0.002100	*Precision@N* @5= 0.081100 @10= 0.09335 @15= 0.13224 *Recall@N* @5= 0.006998 @10= 0.007763 @15= 0.012254	*Precision@N* @5= 0.04470 @10= 0.045000 @15= 0.057000 *Recall@N* @5= 0.001556 @10= 0.0019823 @15= 0.003776	*Precision@N* @5= 0.069971 @10= 0.088221 @15= 0.091223 *Recall@N* @5= 0.003349 @10= 0.003622 @15= 0.004322	*Precision@N* @5= 0.063310 @10= 0.081130 @15= 0.089993 *Recall@N* @5= 0.002331 @10= 0.004112 @15= 0.005211
FST	**FCST**	**FDST**	**FCDT**	**FCDS**
Precision@N @5= 0.070000 @10= 0.075510 @15= 0.089110 *Recall@N* @5= 0.007100 @10= 0.007137 @15= 0.008598	*Precision@N* @5= 0.093117 @10= 0.113750 @15= 0.115520 *Recall@N* @5= 0.008001 @10= 0.008773 @15= 0.009633	*Precision@N* @5= 0.098883 @10= 0.110020 @15= 0.163290 *Recall@N* @5= 0.009921 @10= 0.009631 @15= 0.011260	*Precision@N* @5= 0.083621 @10= 0.11713 @15= 0.18824 *Recall@N* @5= 0.008114 @10= 0.013361 @15= 0.042235	*Precision@N* @5= 0.086640 @10= 0.099910 @15= 0.133710 *Recall@N* @5= 0.007763 @10= 0.009977 @15= 0.011001
FCDST	**[YYLL11]**	**[JSW$^+$12]**	**[WTM13]**	**Fused Matrix Factorization**
Precision@N @5= 0.199120 @10= 0.34181 @15= 0.521000 *Recall@N* @5= 0.030000 @10= 0.03101 @15= 0.031992	*Precision@N* @5= 0.029000 @10= 0.029000 @15= 0.033000 *Recall@N* @5= 0.001277 @10= 0.001190 @15= 0.001198	*Precision@N* @5= 0.403000 @10= 0.405000 @15= 0.419000 *Recall@N* @5= 0.002900 @10= 0.003000 @15= 0.003100	*Precision@N* @5= 0.091000 @10= 0.109100 @15= 0.117900 *Recall@N* @5= 0.001970 @10= 0.001900 @15= 0.002200	*Precision@N* @5= 0.339992 @10= 0.399227 @15= 0.435961 *Recall@N* @5= 0.032054 @10= 0.037000 @15= 0.045000

Table 2.11: Precision@N, Recall@N of different models in Gowalla dataset

2. The frequency-based model had good precision in both datasets but the recall was quite low. This is because it relied on the common visitors to the locations. Besides the common visitors, other contexts also play a major role in the check-in behavior. As this model ignored those contexts, it had many false negatives.

3. The categorical model could not do as expected. This might be due to the avoidance of the spatial context. Though places are of same category,

the farther location would be less likely to be visited. The spatial model also could not perform well. The evaluation shows that the combination of categorical and social or the categorical and spatial gives better result.

4. The social model was found to be best among the models with two contexts. It's performance was better in Gowalla dataset because it is bigger and has lots of friendship relation.

5. It is better to select the social or temporal model if an additional context beyond check-in frequency is to be incorporated.

6. The combination of the spatial and temporal (FDT) contexts was found to be the outperforming among the counterparts. The combination of categorical and social contexts (FCS) were found to be better than the combination of categorical and the spatial (FCD) contexts.

7. The combination of categorical and the temporal contexts (FCT) performed worse than the combination of the social and the temporal contexts (FST).

8. Based on the evaluation, we can see that it is better to select the spatio-temporal (DT) model if we need to incorporate just two contexts. Though the categoric-spatio-social (CDS) slightly outperformed the categoric-spatio-temporal (CDT) in Weeplace dataset, the case was opposite with Gowalla dataset. It is better to select the categoric-spatio-temporal (CDT) or categoric-spatio-social (CDS) model if we need to incorporate just three contexts. The inconsistent performance of FDST among two datasets also indicates that the social relation may not always be a reliable factor for recommendation. This can also be due to the noisy social links (people with non-matching preferences being in a social tie).

9. The category context only works as a good supplement to the other models (FCD performs better than FD, FCST performs better than

FST, FCDS performs better than FDS, and FCDST performs better than FDST) but is not a sole contributor for the good result. So, if we have to opt out an context, then the category context could be the right choice.

10. The FCDST model not only outperformed all of our ranking-based fused models, but also outperformed the relevant fused models proposed in other studies. It also bet the normal matrix factorization-based model (with five latent factors). This is because we had rating matrices of $\sim$98.5% sparsity. FCDST model incorporated more contexts than all of those fused models. This implies that an efficient fusion of the major contexts can improve the recommendation quality.

11. The performance of simple matrix factorization-based model was better when more latent factors were used (we found the model with latent factor 5 performed better than the models with latent factor of 2 and 3). The performance improvement was not that significant with more than 5 factors. The matrix factorization-based fused model (see Eqn. 2.25) performed better with the increasing latent factors (performance with 5 latent factors was better than 2 and 3 latent factors) and also slightly outperformed the FCDST model.

12. The single context model has better execution time because of the simple graph and rank formulation. The FCDST model's better result costs the execution time because unlike other models, it needs to separately compute the spatial and the categorical based ranking to get the unified rank. The computational cost of matrix factorization-based fused model increased with the number of latent factors.

2.6 Conclusion and Future Work

We formulated the personalized POI recommendation using multiple contexts. We also analyzed the impacts of different contexts (the categorical, spatial, social, and temporal) in POI recommendation. We fused different major contexts to get different recommendation models and analyzed the impact of the major contexts. We also fused all the major contexts into a single recommendation model and demonstrated that it can perform better than other fused models. The analysis of the combination of the contexts and the multi-context recommendation model with reasonable performance gain is a novel touch in the relevant area. There are certain limitations of the linearly fused models, for instance, the selected weights of different factors in the linear fusion may be inconsistent across different sets of training, validation, and testing data sets. Inappropriate selection of weights might introduce unnecessary bias in the model. There are many interesting directions to explore, for instance, the analysis of different other factors (for instance, the utility of POI), other datasets, and different other models.

CHAPTER 3

USER ACTIVITY AND LOCATION INFLUENCE ON POINT-OF-INTEREST RECOMMENDATION

The maturity of the smartphone and the World Wide Web (www) technologies have driven many social network applications which have facilitated people to share text and multimedia contents. The social networks that facilitate users to share the check-in (location visit) information are known as the location-based social networks (LBSN)s and provide various information for a recommendation problem that spans beyond the user-location ratings, and comments, for instance the time of the check-in, the category of the Point-of-Interest (POI), the distance of POI from the user's home, the user's friends' visit to that place, and so forth. It's worthwhile to explore and efficiently integrate such information for the desired purpose. A POI recommendation system uses a user's historical check-in information from LBSNs and different contexts to recommend a list of places that are potentially preferable.

Many of the existing POI recommendation systems have focused on either of the temporal (time of the check-in), the geographical/spatial (distance between check-in locations), or the social (friendship, and trust based) contexts. Incorporation of all the major contexts (the categorical, the geographical, the social, and the temporal) of check-ins into a single model is barely explored by other studies. In this paper, we propose a fused model termed GeoTeCS (Geographical Temporal Categorical and Social) for personalized location recommendation. GeoTeCS uses the matrix factorization technique to fuse the major check-in contexts into a recommendation model. The contributions of this chapter are: (i) it proposes a matrix factorization based location recommender that incorporates all the major contexts -the categorical, the geographical, the social, and the temporal contexts into a single model and (ii) it extensively evaluates the proposed model against two real-world datasets Gowalla and Weeplaces. We also present an extension of this study that is focused on modeling the locality-based hierarchical preferences to generate contextual Point-of-Interest sequence recommendation.

3.1 Introduction

The LBSNs, such as Facebook[1], Foursquare[2], Gowalla[3], and so forth have facilitated the users to share their check-in behavior accompanied by the multimedia contents. The analysis of such check-in information has been an interest for effective prediction in the location recommendation domain. Though some success has been achieved using the check-in frequency and the generic recommendation approaches, the better results of recent studies have motivated the community towards the incorporation of the major contexts of the check-in behaviors.

The role of multiple contexts makes the POI recommendation domain special than other domains. Unlike the traditional recommendation problems, the visit frequency can vary across different users and places, resulting in the *sparsity* of the *user-location* frequency matrix. The user's affinity towards the nearby locations adds the constraint of the *spatial context* in this domain.

Although most of our daily activities are highly influenced by our society, its impact on the check-in trends is not always reliable. For instance, the research [YYL10] has shown that only $\sim 96\%$ of people share $< 10\%$ of the commonly visited places and $\sim 87\%$ of people share nothing at all. This unreliability of check-in information diffusion piles up the challenge for the *social context* incorporation. Similarly, the *temporal popularity* (time of the check-in) of a

[1] https://www.facebook.com

[2] https://www.foursquare.com

[3] https://www.gowalla.com

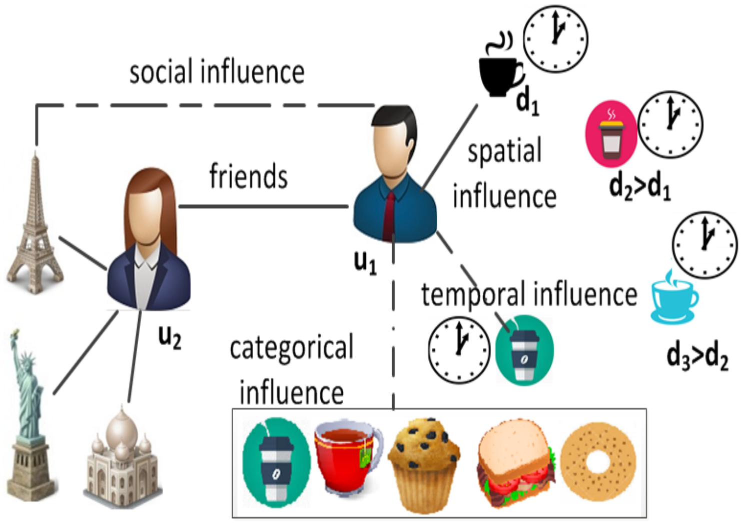


Figure 3.1: Impact of different contexts in check-in trends

place is also another major context. For instance, the bars are more popular in the evenings and the nights. So, relying on just one or two major factors might not be enough for an efficient recommendation.

The Figure-1 illustrates the influence of the categorical, the social, the spatial and the temporal influences in the check-in trend of the users. The figure shows a friendship relation between the users u_1 and u_2. The *social context* can influence the user u_1 to visit the places that were already visited (or recommended) by his friend (u_2). The user (u_1) has check-in(s) at the coffee shop at 1 pm. The *temporal context* may influence the user to visit the same (or other) coffee shop(s) at the same time (of a day). The *categorical context* is reflected if the user visits other places that serve coffee. For instance, most of the shops that serve a breakfast serve the coffee too. The users have preference to the nearest locations [YYLL11]. There are many shops that serve coffee in the afternoon, but the user prefers the nearest one (*spatial influence*) (for instance, the shop at distance d_1 is preferred than the farthest ones (at distance d_2, and d_3)).

There are many other relevant factors, such as, (i) the utility of a POI, (ii) the popularity of the POI (due to the social or other impacts), (iii) the trend of visiting new places, and so forth, which can influence the check-in trend. For instance, users might plan to visit popular places regardless of their distance. The utility of a service is defined in terms of preference of attributes of a service. For instance, if a user is a hiking enthusiast, then she may hike places that are far from her house. The trend to visit new places can influence a user to visit places that might be far, might not have been visited by her friends, and might be of different location type than her past visits. An efficient incorporation of all such major contexts can be challenging as well as beneficial for a good POI recommendation system.

Though the POI recommendation problem is a special area, the techniques used in generic recommendation systems have been explored for POI domain too. For instance, many of them are based on the popular concepts such as, the Collaborative Filtering (CF) [ZCXM09, LSEM12], the Content Based Filtering [YSC$^+$13], and the Hybrid [YZYW13] approaches. Albeit, the POI recommendation is a well explored topic (temporal [YCM$^+$13, JSW$^+$12, WTM13, HJE13, BL16], geographical [YYLL11, BZM12, WTM13, FYL13, HE13, LZX$^+$14, HSL14, ZC15, BL16], social [YYLL11, CYKL12, FYL13, WTM13, ZC15, BL16], categorical [BZM12, HSL14, RW13, LLAM13, ZC15, BL16], sentiment [YZYW13], popularity [RW13, LLAM13]), to our knowledge, the incorporation of all the major contexts (*the categorical, social, spatial, and temporal*) into a single model is not well explored. The main beauty of GeoTeCS is the fusion of all those major contexts into a single efficient recommendation model.

The rest of the chapter is organized as follows: Section 3.2 describes the relevant studies in this area, Section 3.3 describes the methodology of GeoTeCS, Section 3.4 presents the evaluation of our proposed model, Section 3.5 presents the locality-based hierarchical preferences, and Section 3.5.4 presents the concluding remarks of the chapter.

3.2 Related Research

This sections presents the related studies by grouping them according to the approach they used to design their research model.

3.2.1 Simple similarity-based approaches

The spatial context has been defined in Tobler's First Law of Geography [Tob70], ("everything is related to everything else, but the near things are more related than the distant things"). Based on this, Yuan et al. [YCM$^+$13] designed a model with the spatial and the temporal context. They used the cosine similarity measure to identify the users' with similar check-in profiles. They defined the recommendation score for a user-location tuple in terms of the aggregate of the visits count on that location over all the users in the dataset. This was further time constrained by considering only the check-ins that were made in the same location and at the same check-in time. They experimentally claimed that the willingness of a user to visit a location has an inverse relation to the distance from the user's current location. Though their evaluation favored their model, their research didn't address the social, and the categorical contexts.

The social and the spatial contexts were fused in the study from Ye et al. [YYLL11]. They also used the willingness factor and the weighted cosine similarity measure to compare the user profiles for the recommendation. The categorical and the temporal contexts were not explored in their proposed model.

3.2.2 Graph-based approaches

The usage of link analysis has been proposed by Jin et al. [JSW$^+$12] in their personalized PageRank [Hav02] based model. They represented the LBSN as a graph with the users as the nodes, and the users' following/followers link as the directed edges. The model used the personalized PageRank algorithm to compute the rank of the users with respect to a location and a time range. The personalized factor for the (user, location (p), time ($t_1 : t_2$)) tuple was defined as the ratio of the number of check-ins for the tuple to the number of check-ins for the (location (p), time ($t_1 : t_2$)) tuple across all the users. They also used similar approach to define the rank of a location within a time interval. Though they incorporated the temporal context, they ignored the geographical, categorical and social contexts.

Wang et al. [WTM13] defined the problem as a graph with the users and the locations as the graph nodes, the friendship relation as the user-user edges, and the user-location relation as the user-location check-in edges. The friendship based similarity was computed by starting from the target user and by ranking all the users (that formed the user-user link). This was followed by the ranking of all the places visited by those users. The locations with the highest rank value and within a given distance from the past visits of the users were recommended. Their model also ignored the location category context.

3.2.3 Matrix approximation-based approaches

Ding et al. [DJLS07] explored the user-item recommendation problem using the label information propagation. The label propagation is similar to the random walk technique [JS02]. They proposed a learning framework based on the Green's function and applied that to estimate the missing ratings in the user-item rating matrix. In the case of a graph of pairwise similarities, the Green's function can be realized as the inverse of the combinatorial Laplacian. Given the item similarity matrix $\mathbf{W}$, the propagation takes from the labeled data (i.e., items with ratings) to the unlabeled data. The computation of the missing rating was realized as the linear influence propagation. For instance, given the rating from a user as $\mathbf{y}_0^T = (1, 4, ?, ?, ?, ?, 7)$, the estimation of the missing values was made using the influence propagation and was defined as $\mathbf{y} = G\mathbf{y}_0$, where the term G was the Green's function that was obtained from the user-item graph. The rating prediction was then defined as $\mathbf{R}^T = G\mathbf{R}_0^T$, where $\mathbf{R}_0$, is the incomplete user-item rating matrix.

Shao et al. [SWLO09] also used the Green's function as the basis for the linear influence propagation to compute the missing values in the user-music preference matrix for their music recommendation system.

Recently, the matrix factorization models have caught considerable attention due to their scalability and accuracy, which was demonstrated in the seminal research from Koren et al. [KBV09]. Generally, such models learn the low-rank representations (also referred as latent factors) of the users and the items from the user-item rating matrix, which are further used to predict new scores between the users and the items. The non-negative matrix factorization (NMF) approach has attracted the attention of many research areas. Li et al. [LD13] have defined the usage of NMF methods for clustering (for instance, co-clustering, semi-supervised clustering, consensus clustering) and have explained the potential directions of NMF.

Recently, some notable studies in POI recommendation have exploited the fused matrix factorization. Cheng et al. [CYKL12] proposed FMMGM (fused matrix factorization with MultiCenter Gaussian model) that used the Multicenter Gaussian model (MGM) to fuse the geographical and the social contexts of POI recommendation. The MGM relied on the following assumptions: (i) the check-in locations usually clutter around several centers, and (ii) the probability of a user's visit to a location is inversely proportional to the distance from its nearest center. The FMMGM adopted the Gaussian distribution to model the users' check-in behavior. The users' check-ins to a location were sorted based on the check-in frequency and then clustered into centers or regions. All other visited locations within a threshold distance from such centers were considered in the model. If the ratio of the total check-ins (by all the users) in such a region to the total check-ins (from all users to all the places) was greater than a threshold, then those check-ins locations were assumed as a valid region. The likelihood of a user visiting a location was then defined in terms of the aggregated normalized check-in frequency in each center and the normalized probability of the location belonging to that center.

Their fusion framework was a combination of the likelihood of a location belonging to a center (region), and the preference of the user (u) to that location (l). This was defined as: $P_{ul} = P(F_{ul}).P(l \mid C_u)$, where the term $P(F_{ul}) \propto \mathbf{U}_u^T \mathbf{L}_l$ was obtained by using the user topic matrix $\mathbf{U}$ and the location topic matrix $\mathbf{L}$ obtained from the factorization of the *user-location* frequency matrix. Though the experimental results were in favor of the fused social and spatial contexts, the model didn't incorporate the categorical and the temporal contexts.

Lian et al. [LZX$^+$14] proposed the GeoMF which used the factorization model along with the spatial clustering with the two-dimensional kernel density estimation. The locations were divided into grids and the influence of users and locations on those grids were computed. A user's activity or influence area was determined by grid locations $l \in L$ where the user had check-ins. The POI influence area was defined in terms of the collection of locations in grid $l \in L$ to which the influence of this POI could be propagated. The prediction model used the factorized user topic matrix, location topic matrix, user activity matrix and location influence matrix. The fused model was claimed efficient but the impact of categorical, temporal and social contexts remained unexplored.

The GeoMFTD model from Griesner et al. [GAN15] extended the GeoMF model from Lian et al. [LZX$^+$14] to fuse the spatial and the temporal influence but still didn't incorporate other major contexts for the recommendation. For the temporal context, on each POI i, they computed the average time spent by each user to reach the POI j ($j \in g_l$, where g_l is the l^{th} geographical grid/region) from the POI i. This was computed for every user who had at least one check-in at POI i and another (more recent) check-in at the POI j into g_l. The average of such time (t_i^{gl}) for the POI i and all the collocated POIs in the grid g_l for each of the users was computed. The temporal context was addressed by incorporating the temporal coefficients to the POI influence.

Although this model outperformed the traditional ones, it did not incorporate the social and categorical contexts. Furthermore, we think that the check-in time to a POI is as relevant as the time that one spends traveling to that POI or the time that was spent in that particular POI. So, GeoTeCS defines the temporal context as the check-in time to a location and uses this as the basis of the temporal popularity of a location.

3.3 Methodology

The matrix factorization method is one of the most popular methods in the recommender systems. It characterizes both items and users by vectors of factors inferred from the user-item rating matrix. The high correspondence between the item and the user factors leads to a recommendation. The basic idea is to map both the users and the items to a joint latent factor space of dimensionality *f*, which gives the way to model or define the *user-item* interactions in terms of the inner products in that space. The factor matrices are approximated (for instance by using the gradient descent or by other relevant approaches) to have minimal reconstruction error.

Given the user factor matrix $\mathbf{U}$=$[\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_m] \in \mathbb{R}^{lxm}$ and the item factor matrix $\mathbf{V}$= $[\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n] \in \mathbb{R}^{lxn}$, the approximation of the rating matrix $\mathbf{R}$

can be achieved by the multiplication of the low rank factors and can be defined as: $\mathbf{R} \approx \mathbf{U}^T\mathbf{V}$. Due to the sparseness of the rating matrix $\mathbf{R}$, only the observed ratings in the matrix $\mathbf{R}$ can be factorized to define the objective function of the form:

$$\min_{U,V} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (\mathbf{R}_{ij} - \mathbf{U}_i^T \mathbf{V}_j)^2, \tag{3.1}$$

where the term $I_{ij} \in [0, 1]$ is an indicator function where $I_{ij} = 1$ only if the user u_i has a rating for the item v_j. The problem of overfitting can be addressed by regularizing Eqn. (1) as:

$$\min_{\mathbf{U},\ \mathbf{V}} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (\mathbf{R}_{ij} - \mathbf{U}_i^T \mathbf{V}_j)^2 + \frac{\lambda_1}{2} \parallel \mathbf{U} \parallel_F^2 + \frac{\lambda_2}{2} \parallel \mathbf{V} \parallel_F^2, \tag{3.2}$$

where the constants $\lambda_1 > 0, \lambda_2 > 0$ and $\parallel . \parallel_F$ is the Frobenius norm.

According to this concept, each item i is associated with a vector $\mathbf{q}_i \in \mathbb{R}^f$ and each user u is associated with a vector $\mathbf{p}_u \in \mathbb{R}^f$. The resulting dot product $(\mathbf{q}_i^T . \mathbf{p}_u)$ defines the preference of the user u to the item i. This gives the approximation of the user u's rating on the item i, which is denoted by r_{ui}, and the estimate is defined as: $\hat{r}_{ui} = \mathbf{q}_i^T . \mathbf{p}_u$. Often, such model is related to the singular value decomposition (SVD), whose conventional variant is undefined when the knowledge about the matrix is incomplete and is highly prone to over-fitting, if only few known entries are incorporated. Usually, the factor vectors ($\mathbf{p}_\mathrm{u}$ and $\mathbf{q}_\mathrm{i}$) are learned from some objective function by minimizing the regularized squared error on the set of the known ratings. The generic objective function can be defined as:

$$\min_{\mathbf{q},\ \mathbf{p}} \sum_{(u,i) \in k} (r_{ui} - \mathbf{q}_i^T \mathbf{p}_u)^2 + \lambda (\parallel \mathbf{q}_i \parallel^2 + \parallel \mathbf{p}_u \parallel^2), \tag{3.3}$$

where, k is the set of the user-item (u,i) pairs for which the rating/score (r_{ui}) is known and the constant λ is used to control the extent of the regularization.

Many recommendation systems have used the matrix factorization on top of the collaborative filtering because the matrix factorization provides flexibility

Terms	Definition
$\mathbf{R}$	user-location check-in frequency matrix $\mathbf{R}\in \mathbb{R}^{M\times N}$
$\mathbf{P}$	user's latent matrix, $\mathbf{P}\in \mathbb{R}^{M\text{x}K}$
$\mathbf{Q}$	location's latent matrix $\mathbf{Q}\in \mathbb{R}^{N\text{x}K}$
$\mathbf{A}^T$	the transpose of the matrix $\mathbf{A}$
$\Vert\,.\,\Vert_F$	Frobenius norm
g_l	a location grid
F_u	friends of the user u
$r_{u,i}$	rating from user u to item i
$x_{u,i}^{lt}$	activity/influence of the user u in location i at time t, given the grid g_l
P_u	set of locations visited by user u
P_{ut}	set of locations visited by user u at time t
$\mathbb{L}_u$	POIs P_u mapped to the visited areas on the grids;$\mathbb{L}_u \in \mathbb{L}$
y_i^l	influence of the location i to the grid g_l
y_i^{lt}	influence of the location i to the grid g_l at time t
n_u^{lt}	visit frequency of the user u to the grid g_l at time t
σ	the standard deviation
$K(.)$	the standard normal distribution
$d(l_1, l_2)$	geographical distance function between two locations l_1 and l_2
λ	the regularization constant
α, β	tuning parameters

Table 3.1: Terms used in the paper

in terms of bias (for instance, the various data contexts and other application-specific requirements). This facilitates GeoTeCS to incorporate this approach to fuse the major contexts into a single recommendation model.

GeoTeCS is a weighted matrix factorization based model and is inspired from the relevant studies [LZX+14, KBV09, GAN15, Kor10, HKV08]. The incorporation of major contexts makes our model advanced than the studies from Lian et al. [LZX+14] and Griesner et al. [GAN15]. The terms used in this paper are defined in the Table- I. Given a user-location check-in frequency matrix ($\mathbf{R}$) of dimension $M \times N$, it maps the users and the locations into a joint latent space of dimension K $\ll \min(M, N)$ in a way that a user's preference to a location can be defined as the inner product between them in the latent space. The approximation of the frequency matrix can be achieved by solving the following optimization problem:

$$\min_{\mathbf{P},\mathbf{Q}} \parallel \mathbf{R} - \mathbf{P}\mathbf{Q}^T \parallel_F^2, \tag{3.4}$$

where the terms $\mathbf{P}$ and $\mathbf{Q}$ are the user and location latent matrices. The generalization error can be reduced by using the following variant of the optimization function:

$$\min_{\mathbf{P},\mathbf{Q}} \parallel \mathbf{W} \odot (\mathbf{R} - \mathbf{P}\mathbf{Q}^T) \parallel_F^2 + \lambda(\parallel \mathbf{P} \parallel_F^2 + \parallel \mathbf{Q} \parallel_F^2), \tag{3.5}$$

where the Hadamard operator ($\odot$) represents the element wise matrix multiplication and $\mathbf{W}$ is a binary weighted matrix with $w_{ui} \in \{0, 1\}$, and is 1 only if there is at least one check-in by the user u to the location i.

The basic idea behind GeoTeCS is to divide the check-in locations into L grids or regions (g_l such that $\mathbb{L} = \{g_1, g_2, ..., g_L\ \}$). The division can be done either by using the Haversine Formula (which gives the great circle distances between two points using their geo-co-ordinates) or simply by dividing the distance into equal regions (based on the latitude value or based on the density of the check-ins). GeoTeCS realizes the locations as the sequential grids of equal area ensuring each area has location with some check-ins. Along with the two-factor matrices, the users' influence and the POIs' influence are also incorporated into the grids. The user's influence area or activity area is defined as the region/area which depicts high possibility of the appearance of the user. The POI influence area is defined as the popularity of a POI within a grid.

We assume that the influence areas of the POIs have the normal distribution centered at them. The POI influence area is represented by a non-negative vector $\mathbf{y} \in \mathbb{R}_+^L$, where the term y_i^l is the influence of the location i to the grid g_l and is defined as:

$$y_i^l = \frac{1}{\sigma} K(\frac{d(i,l)}{\sigma}), \tag{3.6}$$

where *K(.)* is the standard normal distribution and the term σ is the standard deviation of the distance between the locations in the grid.

There can be some locations with the same category as the location i and still not explored in the past. This may not necessarily indicate the negative preference to this location. As already explained in the Figure 3.1, the locations with the same category might have potential visits. Similarly, if there are some locations in the vicinity that have a check-in time similar to the location i, then their temporal popularity might make them potential POIs too. Such temporal and the categorical bias can be incorporated by extending the POI influence relation (of Eqn. 3.6) and can be defined as:

$$y_i^{lt} = y_i^l + \frac{1}{\mid g_l \mid} \sum_{l' \in g_l} (\mathcal{C}\alpha * y_i^{l'} + \mathcal{T}\beta * y_i^{l'}), \tag{3.7}$$

where $\mathcal{C} \in \{1, 0\}$ and is 1 only if the two locations (*l, l'*) are of the same category, $\mathcal{T} \in \{1, 0\}$ and is 1 only if the check-in time of the two locations are within some threshold ($\triangle T$, we assume the same hour of a day). When none of these is satisfied, we have $y_i^{lt} = y_i^l$ (*only the spatial context*). The terms α and β are tuning parameters. This relation defines the integration of the categorical and the temporal context in the popularity of a location. The location's influence area can then be defined in terms of a non-negative vector $\mathbf{y} \in \mathbb{R}_+^L$, where the term $y_i^{l,t} \in \mathbf{y}$ is the influence of a location i at the time t, to the location grid $g_l \in \mathbb{L}$.

Similarly, the activity of a user in a given location can be defined using the location grids. The basic idea is to dissipate the check-in history among the grids and to find the activity of the user in those grids. The estimated density of a user u at a POI i can be defined as:

$$\frac{1}{\mid P_u \mid \sigma} \sum_{j \in P_u} K(\frac{d(i,j)}{\sigma}), \tag{3.8}$$

where P_u is the set of locations visited by the user u and the σ is the standard deviation of the distances previously visited by the user.

The user's activity can then be defined in terms of a non-negative vector $\mathbf{x} \in \mathbb{R}_+^L$, where the term $x_{u,i}^{l,t} \in \mathbf{x}$ is the influence of a user u to the location i at the time t, with respect to the locations belonging to the grid/region $g_l \in \mathbb{L}$.

As the user's visit is influenced by the social context, we integrate the influence of all the friends while computing the influence of a user. The user's activity vector $\mathbf{x}$ can then be defined as:

$$x_{u,i}^t = \frac{1}{\mid P_{ut} \mid} \sum_{l \in \mathbb{L}_u} \frac{n_u^{lt}}{\sigma} K(\frac{d(i,l)}{\sigma}) + \sum_{u' \in F_u} \frac{1}{\mid P_{u't} \mid} \sum_{l' \in \mathbb{L}'_u} \frac{n_{u'}^{lt}}{\sigma'} K(\frac{d(i,l')}{\sigma'}). \tag{3.9}$$

Using the POI influence area and the user's influence area, the optimization problem can be redefined as:

$$\min_{\mathbf{P},\mathbf{Q},\ \mathbf{X}} \parallel \mathbf{W} \odot (\mathbf{R} - \mathbf{P}\mathbf{Q}^T) - \mathbf{X}\mathbf{Y}^T \parallel_F^2 + \lambda(\parallel \mathbf{P} \parallel_F^2 + \parallel \mathbf{Q} \parallel_F^2) + \gamma \parallel \mathbf{X} \parallel_F^2 . \tag{3.10}$$

The term γ is used to control the sparsity across the user-location-grids. The dimension of $\mathbf{X}$ and $\mathbf{Y}$ matrices is dependent on the number of location grids L $\ll \min(M, N)$, so we have $\mathbf{X} \in \mathbb{R}^{M \times L}$ and $\mathbf{Y} \in \mathbb{R}^{N \times L}$. We have $\mid T \mid$ copies of $\mathbf{X}$ and $\mathbf{Y}$ matrices, where each copy represents one of the time slot $t \in T$.

The preference matrix can then be defined by integrating these factor matrices and can be defined as:

$$\hat{\mathbf{R}} = \mathbf{P}\mathbf{Q}^T + \mathbf{X}\mathbf{Y}^T, \tag{3.11}$$

where **P** and **Q** are the user topic and the location topic matrices, and **X** and **Y** are the user's activity and the location influence matrices respectively.

Using the factorized matrices **P**, **Q** and the influential matrices **X** and **Y**, the estimated preference of a user u, to the location i at the time t is then defined as:

$$p_{u,i,t} = \mathbf{P}_u\mathbf{Q}_i^T + \mathbf{X}_{u,t}\mathbf{Y}_{i,t}^T, \tag{3.12}$$

3.4 Evaluation

This section presents the details of the dataset, the metrics used for evaluation, and the findings and detailed discussions on them.

3.4.1 DataSet

The Weeplaces and the Gowalla dataset [LLAM13], which was collected from the popular LBSNs Gowalla and Weeplaces was used for evaluation. These datasets are well defined and had all the attributes (the location category, the geo-spatial co-ordinates, the friendship information, and the check-in time) relevant to the model. The incomplete records were eliminated in the evaluation. The statistics of the dataset is defined in the Table 3.2. The Gowalla dataset had only 7 main location categories, so we used the well defined subcategories instead.

The 5 most checked-in location categories are listed in Table 3.3. The work or home related category "Home / Work / Other: Corporate / Office" was popular from 6 am to 6 pm, with the highest check-ins (42,019) made at 1

Attributes	**Gowalla**	**Weeplaces**
Checkins	36,001,959	7,658,368
Users	319,063	15,799
Locations	2,844,076	971,309
Social links (*undirected*)	337,545	59,970
Location Categories	629	96

Table 3.2: Statistics of the dataset

Gowalla	**Weeplaces**
Corporate Office (1,750,707)	Home / Work / Other: Corporate / Office (437,824)
Coffee Shop (1,063,961)	Home / Work / Other:Home (306,126)
Mall (958,285)	Food:Coffee Shop (267,589)
Grocery (884,557)	Nightlife:Bar (248,565)
Gas & Automotive (863,199)	Shops:Food & Drink:Grocery Supermarket (161,016)

Table 3.3: Top -5 visited location categories (and their check-ins count)

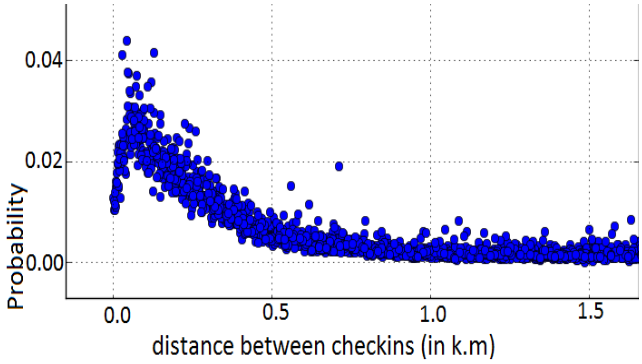


Figure 3.2: Impact of distance to check-in trend in Weeplaces dataset

Models	Precision	Recall	F-Score
Ye et al. [YYLL11]	0.02417	0.00095	0.00183
LBSNRank [JSW+12]	0.08496	0.00063	0.00125
Wang et al. [WTM13]	0.01818	0.00052	0.00100
FMFMGM	0.06549	0.00487	0.00906
GeoMFTD	0.09415	0.00676	0.01261
GeoTeCS	0.29800	0.01546	**0.02939***

Table 3.4: Average Performance of GeoTeCS and other models in Weeplace dataset

pm. Similarly, the **bars** had highest of 21,806 check-ins at 2 am and the lowest check-ins (15,209) at 5 am. Most of the check-ins were at 12 pm to 6 pm and were in either home or work related categories.

We also analyzed the impact of distance on the check-in behavior. For every user, the check-ins were chronologically sorted and the distance between consecutive check-ins of each user was computed. The likelihood of a user to check-in at particular distance (for convenience, the distance was arbitrarily rounded to four decimals) was estimated by her visit history. Figure 3.2 illustrates the inverse relation of check-in trend to the distance of the POI in Weeplaces dataset[4]. We can see that most of the users' check-ins are centralized within some distance (the dense patches within 0.5 km indicate that most of the users' had the check-ins in the near places). The figure shows that the willingness of check-in decreases with the increasing distance of the location.

3.4.2 Results

GeoTeCS was evaluated using 5-fold cross-validation. The precision (P), the recall (R) and the F-score (2*P*R/(P+R))) metrics for the top N recommended items (we considered four cases, (i) top 5, (ii) top 10, (iii) top 15, and (iv) top 20 items with the highest recommendation score) were used. The process was repeated with three sets of values for α:β (0.25:0.75, 0.5:0.5, 0.75:0.25).

[4] though the trend on Gowalla dataset is not shown, it also had similar trend

Models	Precision	Recall	F-Score
Ye et al. [YYLL11]	0.03000	0.00120	0.00230
LBSNRank [JSW$^+$12]	0.40900	0.00300	0.00600
Wang et al. [WTM13]	0.10600	0.00200	0.00392
FMFMGM	0.07220	0.00800	0.01440
GeoMFTD	0.09900	0.01570	0.02710
GeoTeCS	0.38477	0.03410	**0.06264***

Table 3.5: Average Performance of GeoTeCS and other models in Gowalla dataset

Models	Precision@N	Recall@N
Ye et al. [YYLL11]	@5= 0.03030 @10= 0.02300 @15= 0.01910	@5= 0.00080 @10= 0.00090 @15= 0.00100
LBSNRank [JSW$^+$12]	@5= 0.08530 @10= 0.08480 @15= 0.40900	@5= 0.00060 @10= 0.00060 @15= 0.00300
Wang et al. [WTM13]	@5= 0.04490 @10= 0.04140 @15= 0.04070	@5= 0.00140 @10= 0.00207 @15= 0.00220
FMFMGM	@5= 0.05900 @10= 0.06800 @15= 0.08700	@5= 0.00489 @10= 0.00687 @15= 0.00873
GeoMFTD	@5= 0.07719 @10= 0.08947 @15= 0.11578	@5= 0.00641 @10= 0.00824 @15= 0.00924
GeoTeCS	@5= **0.28400** @10= **0.36500** @15= **0.38800**	@5= **0.00950** @10= **0.00920** @15= **0.02770**

Table 3.6: Precision@N, Recall@N of GeoTeCS against other studies

When computing the POI influence region (refer Eqn. (7)), the best result was achieved when the categorical factor (α) was set to 0.25 and the temporal factor (β) was set to 0.75. The hourly time slot was used to compare the check-in hours. We compared the performance of the following fused models: (i) model from Ye et al. [YYLL11], (ii) LBSNRank [JSW$^+$12] (iii) the model from Wang et al. [WTM13], (iv) FMMGM, (v) GeoMFTD, and (vi) GeoTeCS. The comparative performance of the different models is illustrated in the Table 3.4 and Table 3.5 The comparison of average precision, recall measure across top 5, 10, 15 recommendation scores for Weeplaces dataset is illustrated in Table 3.6. From the evaluation, we can see that GeoTeCS consistently outperforms the relevant models. Based on this evaluation, we claim that the efficient integration of the major contexts of check-in behavior results in a more accurate recommendation.

3.5 Locality-based influence on hierarchical preferences

The POI preference of a user varies by locality, item type, and the co-visitors, e.g., user$_1$ and user$_2$ can have closest preference on *food* items but not on *historic* sites, etc. A locality can have different preference trends (e.g., popular for food, recreation, etc.) and a user's preference can span across multiple such trends. A good recommender should also exploit the aggregated locality preference trends. Most of the existing studies group items by category or global user preferences which might not be relevant for locality-based aggregated preferences. We propose a model termed as HiRecS (**Hi**erarchical Contextual Location **Rec**ommendation **S**ystem) that formulates user preferences as hierarchical structure and presents a hierarchy aggregation technique for POI recommendation. The top level of locality hierarchy contains preferred k items from a set of users and the subsequent levels contain preference wise subsets. The core contributions of this research are: (i) it formulates user preferences as a preference hierarchy, presents a technique to aggregate preference hierarchies of a similar users, and models the target users' preference in terms of aggregated trend in a locality, (ii) it contextually exploits the aggregated trend to generate personalized POI sequences, and (iii) it extensively evaluates the proposed model with two real-world datasets and demonstrates performance gain (0.03 - 0.28 on pair F-score, 0.006 - 5.91 on diversity, 0.0349 - 17.51 on displacement, and 0.114 - 0.289 on NDCG) over baseline models.

The POI (also referred as location) recommenders exploit explicit check-in information and some implicit contexts [YCM$^+$13, ZC15, BL16, XNL$^+$17, XLLZ17] to generate a list of POIs that are relevant to user preferences. Generally, most of the POI recommenders recommend a single POI or a list of POIs [YCM$^+$13, ZC15, BWLC16, BL16, BL17, AK17] that satisfy the personalized user preferences. However, the user preferences are contextually dynamic, for instance a user might prefer historical sites when she visits one locality and might prefer religious sites in another locality, similarly she might prefer one set of POIs when she is with family and another set of POIs when with her friends. Figure 3.3 illustrates the dynamic preferences of users. The user u_1 has the closest match with u_2 and u_3 for *religious* sites (item type1), with u_2 and u_6 for *food* (item type3). The user u_2 has no match with u_1 for *recreational* sites (item type2). Such a locality-based contextually dynamic preferences are not easily captured if we model preferences globally.

We address the contextual and locality preferences by modeling user preferences based on regions they have visited. The exploitation of such a contextual and locality-based influence on user preferences is still a viable problem. Traditional systems design categorical hierarchy of items or globally categorize users and items based on the consumption experience. However, the extensive hierarchical relation can be difficult to obtain, complex in structure (e.g., it is not only the categorical attribute that defines user preferences and the global preference of users can have many constraints and patterns which may not be represented by simple hierarchies, on the other hand locality-based preference hierarchies are smaller, confined to a locality, and are simple in structure), difficult to model (e.g., it is difficult to handle extensive preference hierarchies in efficient and scalable manner), and computationally expensive.

Unlike the traditional categorical hierarchy of items or global consumption-based extensive hierarchy, we represent the user preferences as locality-based hierarchical structures, where the preferences of users are modeled on each locality/region and are represented as hierarchical structures. We define the contextual preference-based hierarchical structure, and exploit it to generate POI sequence recommendation. The top level of hierarchy contains the k preferred items of a set of users and the lower levels contain preference wise subsets (e.g., the top level can contain a set of preferred items to a user, the second level can distinguish the items by social context, i.e. preferable for visit with friends or preferable for visit with family, etc.). We perform region/locality-based separation of hierarchy to represent a semantically coherent set of POIs that have a similar trend of user preferences. For a target user, the closest matching k clusters of former visitors in a locality are discovered. The hierarchical preferences of each cluster are generated. As a user's preferences can overlap across multiple sets of preferences, we define a hierarchy aggregation technique to aggregate the preference hierarchies of top-k clusters that are similar to the target user. The aggregated hierarchy is contextually traversed to generate the recommendation.

The main contributions of this research are: (i) it models the user locality preferences as hierarchical structure, (ii) it presents a hierarchy aggregation technique to model the aggregated preferences of multiple sets of users in a locality and contextually exploits the aggregated hierarchy for POI sequence recommendation, and (iii) it demonstrates the efficiency of proposed model using pair F-score, diversity, displacement, and NDCG metrics on two real-world datasets.

The contextual preference hierarchy formulated by our model can be of potential interest for many real-world applications, such as:

1. Personalization: The contextually aggregated hierarchy represents the personal preferences of users. The hierarchy can be traversed to track the personalized preferences at different levels (e.g., the top level of the

hierarchy can represent preference on "Food", the lower level can represent preference on specific food item).

2. Preference-based association mining: We can define the association of items which match the preferences of similar users at different levels of hierarchy. The hierarchy can be exploited to mine association rules [BL97, HPK11]. The top level of hierarchy can be used to extract generic rules, such as: "20% of users who visited a *Restaurant* are most likely to visit a *Coffee Shop*", and the lower level of hierarchy can be used to extract specific rules, such as "10% of users who visited "*Townhouse Grill*" (a specific restaurant) are most likely to visit the "*Starbucks*" (a specific coffee shop)".

3. Question answering: The hierarchical structure can be adapted to a question answering system in an interactive environment. The initial input can be applied to the root of the hierarchy and the interaction can proceed by matching the user inputs with the levels of hierarchy.

4. Recommendation System: The preference-based hierarchy can be exploited for contextual and personalized recommendation system. The contexts can be applied to the hierarchy in order to reach the best matching leaf node and to find the best item. Our study presents the application of contextual hierarchy to generate POI sequence recommendation.

5. Knowledge Discovery: The preference-based relation between user-user, user-item, and item-item can be extracted by comparing their relevant hierarchies.

6. Clustering of POIs and users: The common preferences of users and items extracted from the hierarchy can be used to cluster users and items. For instance, cluster of users who prefer recreational sites, cluster of users

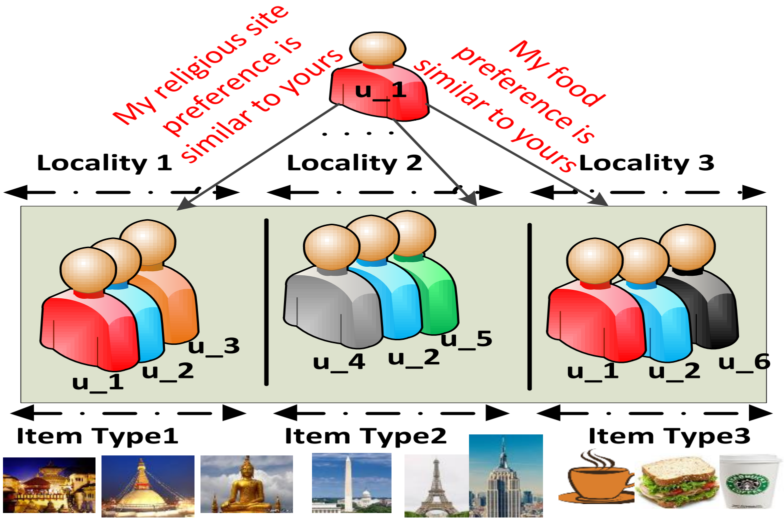


Figure 3.3: User locality preferences

who prefer religious sites, cluster of POIs popular in evening, cluster of POIs popular for food, etc.

7. Preference-based comparison: The hierarchical structure can also facilitate comparison of consumption and preference information of different users. For instance, the check-in information of users can be represented as trees which can be used to compare and analyze the consumption experience of users.

Most of the existing studies exploited collaborative filtering [YXYG16], apriori principle [YXYG16], topic-modeling [LCX+14], tree-based [ZLWS15], matrix factorization [GLX+11], and neural networks [BILB18]. The model from Wang et al. [WLC+16] handled crowd constraints (e.g., peak hours of POIs) by extending the Ant Colony Optimization algorithm. The ranking model [JQMF16] personalized travel sequences in different seasons by merging textual data and viewpoint information extracted from images but ignored social and temporal preferences. Lim et al. [LCLK17] exploited geo-tagged images and contexts, such as visit duration, users' preferences, and start/end POIs to define time-based user preferences but ignored the categorical, temporal, and social constraints. A probabilistic model [COX16] used Rank-SVM to rank the items and used Markov model to predict the transition between POIs. Most of the existing studies have exploited few contexts and have focused on personalized POI visit durations. To the best of our knowledge, none of the previous studies have exploited locality-based hierarchical preference aggregation for contextual POI sequence generation.

3.5.1 Methodology

This section defines the relevant preliminary concepts and the proposed model.

Preliminaries

In this subsection, we define the preliminary concepts used in this paper.

1. Context: A context (e.g., current time, POI distance, etc.) of a check-in represents the current and previous scenarios which have (in)direct influence on the selection of next POI and can be represented as a high dimensional vector.

2. Context-aware POI sequence: Given a set of contexts C $= \{c_1, c_2, ..., c_i\}$, our objective is to predict a sequence of POIs relevant to the given context and user preferences. For a user u, we define the travel history as an ordered sequence $H_u = (V_1, V_2, ..., V_n)$, where the check-in triplet $V_i = (l_i, a_i, d_i)$ indicates the location (l_i), arrival time (a_i) to l_i, and the departure time (d_i) from the location l_i.

$$
\begin{aligned}
ST(i) &= \frac{1}{\mid U \mid} \sum_{\substack{u \in U \\ V_u \in H_u \\ V_u.l = i}} \frac{1}{\mid V_{u,i} \mid} \sum_{l \in V_{u,i}} (d_l - a_l) \\
&= \frac{1}{\mid U \mid} \sum_{\substack{u \in U \\ V_u \in H_u \\ V_u.l = i}} \frac{1}{\mid V_{u,i} \mid} \sum_{l \in V_{u,i}} (a_{l+1} - TT(l, l+1) - a_l).
\end{aligned} \tag{3.13}
$$

3. Visit duration of POI: The stay time or visit duration of a POI is defined by the time spent on the POI. The average visit stay time (ST) of a POI i is the average time spent by all visitors and is defined as in Eqn. 3.13. The term U is the set of all users, $V_{u,l}$ is the set of visits made by the user u to location l, TT(a,b) is the travel time between POI a and POI b. We use a log-normal distribution to compute the travel time between two POIs visited consecutively. The stay time is [0,1] normalized using min-max normalization and is represented as $ST'(i)$.

   We define the user interest to a place in terms of an aggregate of stay time (AST) to that place. This term, in turn, relies on the visit frequency, stay time to that place, and the stay time to the places of the same category:

$$
AST(u,i)_{cat} = (1 - \alpha) * \mathcal{S}(u,i) + \alpha * \mathcal{C}(u,i) \sum_{\substack{l \in V_u \\ l.cat = i.cat}} \frac{ST'(l)}{V'_{u,l}}, \tag{3.14}
$$

   where V_u is the set of visits by user u, $l.cat$ is the category of location l, $V_{u,l}$ is the set of visits by user u to location l, $V'_{u,l} = \frac{|V_{u,l}|}{|V_u|}$ is the normalized visit count of user u to location l, and $0 \leq \alpha \leq 1$ is a constant tuning factor used to balance the impact of stay time on a place and that of the places with same category, and can be obtained by using the fraction of check-ins that are of same category as location i (see Sec. 3.5.3 for more detail). The term $\mathcal{S}(u,i) = \frac{ST'(i)}{V'_{u,i}}$, if $\mid V_{u,i} \mid > 0$ and $\mathcal{S}(u,i) = 0$ otherwise, is the contribution of historical check-ins. The term $\mathcal{C}(u,i) = \frac{1}{\sum_{\substack{l \in V_u \\ l.cat = i.cat}} 1}$, if u has check-ins on category $i.cat$ and $\mathcal{C}(u,i) = 0$ otherwise is the categorical contribution to the stay time. Similarly, using the social impact, the average stay time on a location i can be defined as:

$$AST(u,i) = (1-\psi_1) * AST(u,i)_{cat} + \psi_1 * \mathcal{G}(F_u) \sum_{k \in F_u} AST(k,i)_{cat},$$
$$\mathcal{G}(F_u) = \frac{1}{\mid F_u \mid} \text{ if } \mid F_u \mid > 0,\ \mathcal{G}(F_u) = 0, \text{ otherwise,} \tag{3.15}$$

where F_u denotes the set of friends of user u, $0 \leq \psi_1 \leq 1$ is a tuning factor to model the social impact, and can be estimated using the fraction of check-ins of the user u that are common to her friends (see Sec. 3.5.3 for further detail). The average stay time by a user u to a location category *'cat'* can be defined as:

$$AST(u)_{cat} = (1-\gamma_1) * (\sum_{\substack{i \in V_u \\ i.cat = cat}} AST(u,i)_{cat}) + \gamma_1 * (\sum_{\substack{j \in F_u \\ k \in V_j \\ k.cat = cat}} AST(j,k)_{cat}), \tag{3.16}$$

where $0 \leq \gamma_1 \leq 1$ is a tuning factor estimated using the fraction of check-ins of user u that are common to her friends and have category *'cat'*. AST_{cat} is the aggregate of average stay on the category *'cat'* from all users and AST^t_{cat} gives the measure for time t.

4. Preference score of POI: Given a user u, her preference score (PS) for a place l at time t is composed of the historical check-ins, categorical contribution, and the average stay time:

$$PS(u,l,t) = \beta * \{(1-\theta) * \mathcal{P}(u,l) * \mid V_{u,l,t} \mid + \theta * \mathcal{Q}(u,l) \sum_{\substack{l' \in L \\ l'.cat = l.cat}} \frac{\mid V_{u,l',t} \mid}{\mid V_{u,l'} \mid}\}$$
$$+ (1-\beta) * AST(u,l), \tag{3.17}$$

where $V_{u,l,t}$ is the set of visits made by user u to location l at time t, L is the set of all locations, $0 \leq \theta \leq 1$ can be estimated as in Eqn. 3.14, and $0 \leq \beta \leq 1$ is a tuning factor (see Sec. 3.5.3 for detail). The term $\mathcal{P}(u,l) = \frac{1}{|V_{u,l}|}$, if $| V_{u,l} |> 0$ and $\mathcal{P}(u,l) = 0$ otherwise, is the contribution from historical check-ins. The term $\mathcal{Q}(u,l) = \frac{1}{\sum\limits_{\substack{l \in V_u \\ l.cat = i.cat}} 1}$, if $\exists V_{u,l} \wedge l.cat = i.cat$ and $\mathcal{Q}(u,l) = 0$ otherwise, is the categorical contribution. This relation addresses the trade-offs between visit frequency and stay time, which is crucial to reward the preferred check-ins with low frequency but reasonable stay time. The relations defined above incorporate categorical and social contexts and can handle the cold-start items (items with no check-ins) and cold-start users (users with no check-ins) to some extent. The generalized preference $PS(l,t)$ is derived from above relation by considering the visit frequencies and stay time of all the visitors of this location at time t. A consolidated preference score is defined to address the trade-off between constraints (e.g., travel time, distance, etc.) and preference score:

$$P(u,l,t) = PS(u,l,t) * (1 - \frac{1}{m}\sum_{i=1}^{m} Constraint_i(l,p)), \tag{3.18}$$

where $Constraint_i(l,p)$ is a normalized numeric measure of i^{th} constraint between the users' current location p and the target location l. For instance, the spatial constraint is the measure of distance between locations p and l which is min-max normalized by using the distance traveled by any user to reach location l from any other location.

3.5.2 System architecture

Figure 3.4 shows the block diagram and Figure 3.5 shows the high-level overview of the proposed model. The core functionalities of the model are:

1. Location profile creation: A location profile is a concatenation of the category vector $\langle cat_1, cat_2, ..., cat_i \rangle$, distance vector $\langle dist_1, dist_2, ..., dist_j \rangle$,

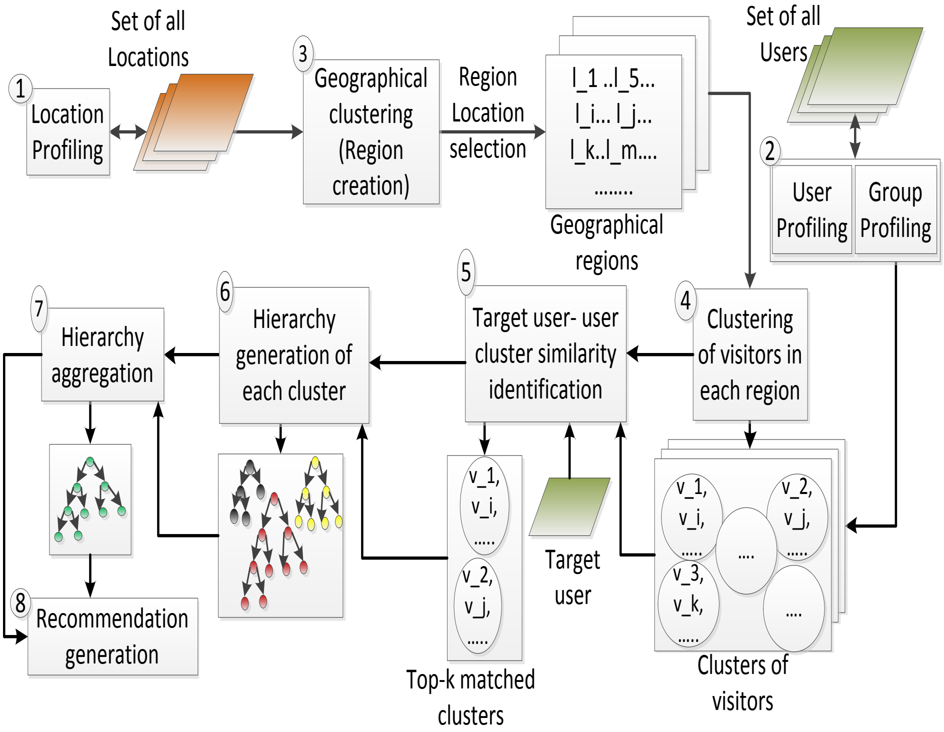


Figure 3.4: Block diagram of the proposed model

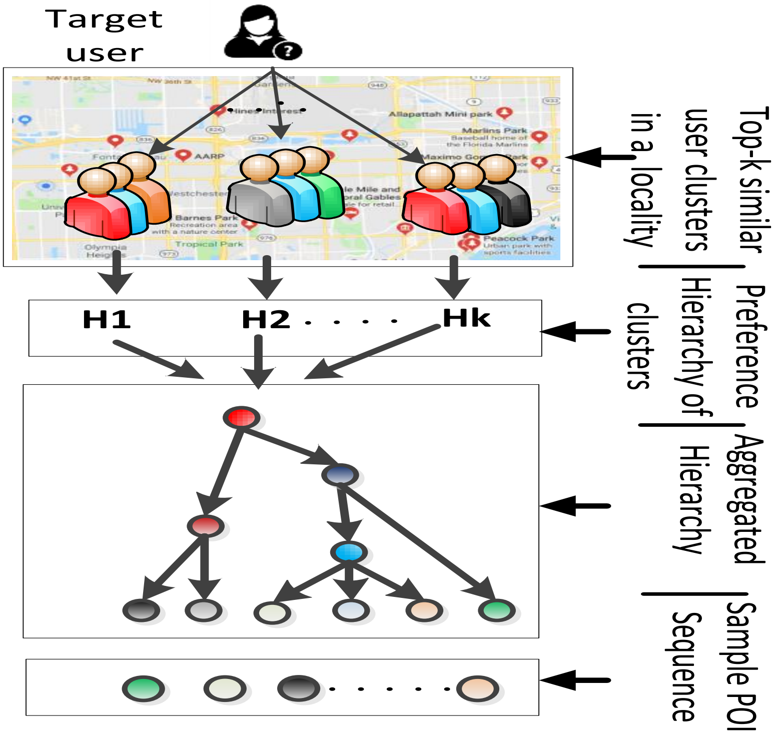

Target user
H1
H2 . . . . Hk
Top-k similar user clusters in a locality
Preference Hierarchy of clusters
Aggregated Hierarchy
Sample POI Sequence

Figure 3.5: High level overview of the proposed model

and time vector $\langle time_1, time_2, ..., time_k \rangle$. The vectors' value is defined by the vector type, e.g., the index of element cat_1 is set to 1(0) if the place is (is not) of category cat_1, index of $time_1$ has the frequency of check-ins made at time $time_1$ on this place, and index of $dist_1$ has the number of check-ins to this place when the previous place was $dist_1$ distance far from it. The vectors are normalized (see Sec. 2a for normalization approach) before concatenation. We use hourly variants for time and five variants (i.e., 1 Km, 2 Km, 5 Km, 10 Km, and more than 10 Km) for distance.

2. User Profile creation: The user profiles are created using the historical check-in information of the users.

   (a) Single user profile creation: A user profile is a concatenation of feature vectors as in the location profile. It contains the relevant frequency, for e.g., the term cat_1 represents the number of user check-ins to category cat_1, etc. The individual features are weighted according to user preferences. For e.g., a user might have more affinity for the *distance* of places rather than *price*, etc. We use the check-in counts and the frequency counts of a feature to calculate the preference of features. The preference of a user on feature f_i is defined as $pref(u, f_i) = \frac{|V_u|}{\phi(u,f_i)}$, where $\mid V_u \mid$ is the total visits made by the user u and $\phi(u, f_i)$ is the count of unique feature f_i from all visits of u. For e.g., if all of the 100 check-ins of a user are made to 100 different categories, then there is no repetition of the category and the categorical preference is $100/100 = 1$. However, if the check-ins are on 50 different categories, then some categories are preferred and repeated, hence the user has some categorical preference (here $100/50 = 2$). As the variants for a feature might be different (for e.g., there might be 20 variants for place category, 24 variants (i.e. hourly) for check-in time, 4 variants (single, family, friends, couple) for social feature, etc.), the preference-weighted feature vectors are normalized (we use min-max normalization) before concatenation.

(b) User group profile creation: The user group/cluster profile is an aggregation of all users' profile in the group. It is an aggregated preference of its members on all the features and represents the preferences of the group. As we use soft clustering of users, the preference of a user to a group/cluster should be taken into account. Given a group $G = \{(u_1, w_1), (u_2, w_2), ..., (u_m, w_m)\}$, where each pair represents a user's profile and the preference of user to the group G, the aggregated profile $\mathcal{P}(G)$ is defined as:

$$\mathcal{P}(G) = \frac{1}{m}(\frac{w_1}{w}u_1 + \frac{w_2}{w}u_2 + ... + \frac{w_m}{w}u_m), \tag{3.19}$$

where w_i is the fraction of check-ins from u_i that contribute to the group $\mathcal{G}$, and $w = \sum_{i=1}^{m} w_i$. As user preferences vary by regions, we define user clusters for each region and the preferences of users on the regions are incorporated accordingly.

3. Geographical clustering: Inspired from the relevant studies [LZX+14, BILZ18, BWLC16, BIM19], we divide the check-in locations into L uniform grids such that $\mathbb{L} = \{g_1, g_2, ..., g_{\mathrm{L}}\}$. We use Haversine Formula (it gives the great circle distances between two points using their geo-coordinates) to create regions which can contain overlapping sets of locations.

4. Clustering of visitors in each region: The visitors of each region are clustered to represent the users with similar check-in preferences and most likely with similar preference on order of features. We use soft clustering to incorporate dynamic user preferences which are not captured by a single cluster. We adopt the Gaussian mixture model to define a

probabilistic model for cluster membership of each object x_i as:

$$p(x_i \mid K) = \sum_{k=1}^{K} \pi_k g_k(x_i), \tag{3.20}$$

where the term $g_k(x_i) = \mathcal{N}(x_i \mid \mu_k, \text{Cov}_k)$ is the Gaussian distribution with mean μ_k and covariance matrix Cov_k, π_k is the weight of k^{th} mixture component, $\sum\limits_k \pi_k = 1$, and K is the number of clusters. Each of the Gaussian distribution component represents a locality of user activity, and the mean value denotes the latitude and longitude of the locality center. The centers can be user's home, office, or her favorite place. The parameters of the model and the membership can be determined by maximizing the following relation:

$$l(K) = \sum_{i=1}^{n} log(p(x_i \mid K)) \tag{3.21}$$

We use the Expectation-Maximization (EM) algorithm to achieve the above objective. The *Expectation* step evaluates the responsibility using initial parameters as:

$$\gamma_i^k = \frac{\pi_k g_k(x_i)}{\sum\limits_{k=1}^{K} \pi_k g_k(x_i)}, \tag{3.22}$$

where γ_i^k is the responsibility of item x_i to cluster k and $i = 1, 2, ..., n$.

The *Maximization* step re-estimates parameters using the responsibilities computed in expectation step:

$$\begin{aligned} \mu'_k &= \frac{1}{n_k} \sum_{i=1}^{n} \gamma_i^k x_i, \\ \text{Cov}'_k &= \frac{1}{n_k} \sum_{i=1}^{n} \gamma_i^k (x_i - \mu_{MLE})(x_i - \mu_{MLE})^T, \\ \pi'_k &= \frac{n_k}{n}, \end{aligned} \tag{3.23}$$

where $n_k = \sum\limits_{i=1}^{n} \gamma_i^k$, $\mu_{MLE} = \frac{1}{n} \sum\limits_{i=1}^{n} x_i$, and we repeat until convergence.

5. Target user and user cluster similarity: The user clusters (see Sec. 2b) are used to capture the location preferences of users. For a target user who visits a region, the top-k matching clusters' profiles are used to represent her preferences and can be used to recommend relevant places in the region. A cosine similarity is intuitively used to find the top-k matching clusters for a target user u_j.

$$similarity(u_j, C_k) = \frac{\sum_{i=1}^{n} u_{j,i} * C_{k,i}}{\sqrt{\sum_{i=1}^{n} u_{j,i}^2}\sqrt{\sum_{i=1}^{n} C_{k,i}^2}}, \tag{3.24}$$

where $C_{k,i}$ is the i^{th} term from the vector for cluster C_k and $u_{j,i}$ is the i^{th} term from the profile of user u_j.

6. Hierarchy generation for each user cluster: The hierarchical structures of the top-k matching clusters are generated to model the hierarchical preferences of users in the clusters. For each user cluster, the preference is hierarchically defined using the set of places from the target region. Inspired from [ZLHL13], we use the conditional mutual information (CMI) [CT12] metric to generate the hierarchy. For every two places X and Y and a cluster C_i, the CMI metric gives the expected value of the mutual information of X and Y on the cluster C_i and is defined as:

$$CMI(X;Y|C_i) = H(X, C_i) + H(Y, C_i) - H(X, Y, C_i) - H(C_i), \tag{3.25}$$

where the function $H(.)$ denotes an entropy. $H(X, C_i)$ is defined in terms of the fraction of check-ins to POI X that are contributed by members of C_i. We use:

$$p(X, C_i) = \frac{\alpha + \sum_{u \in C_i} \mid V_{u,X} \mid}{\alpha * N + \mid V_X \mid}$$

and

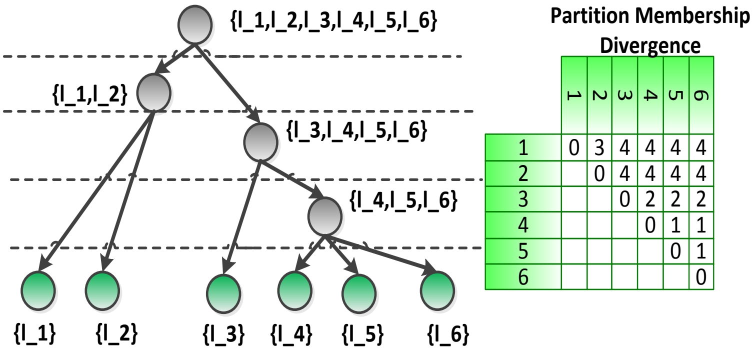
$$H(X, C_i) = -p(X, C_i) \log_2(p(X, C_i)), \tag{3.26}$$

where V_X is the number of check-ins made to POI X, $\sum\limits_{u \in C_i} \mid V_{u,X} \mid$ is the number of check-ins made by members of cluster C_i to the POI X, α>0 is a smoothing factor (used to simplify the computations for POIs with no check-ins), and N is the total number of users. The term $H(Y, C_i)$ is defined accordingly. Similarly, the term $H(X, Y, C_i)$ is defined by the fraction of check-ins from users of cluster C_i who have visited both POIs X and Y. The term $H(C_i)$ is defined in terms of the number of user clusters to be used:

$$H(C_i) = -p(C_i) \log_2(p(C_i)), \tag{3.27}$$

where $p(C_i) = \frac{1}{K}$ and K is the number of user clusters to be used. This gives the CMI matrix for each cluster of users.

It is to be noted that the simple similarity measure between places X and Y is always same on all user clusters and is unable to model contextually dynamic preferences of different user clusters. The similarity between places is dependent on the visitors of a locality (e.g., for users u_1 and u_2 places l_1 and l_2 might be semantically similar but not for users u_2 and u_3) and should be modeled accordingly. The CMI metric facilitates us to incorporate the preference of user clusters on any pair of POIs X and Y. For a given cluster of users, its CMI metric matrix can be transformed into region/locality specific places similarity matrix by setting the diagonal entries to 1 and normalizing other entries. We use this similarity matrix and the complete link clustering to get a user cluster hierarchy because it is less susceptible to noise and outliers [TSK06]. The obtained hierarchy may not explicitly split places using mutually explicit criteria, but performs the splitting based on their entropy. Given a set of places in a region, a hierarchy represents the check-in preference for a cluster of users, and each node of the hierarchy represents some implicit preference association.

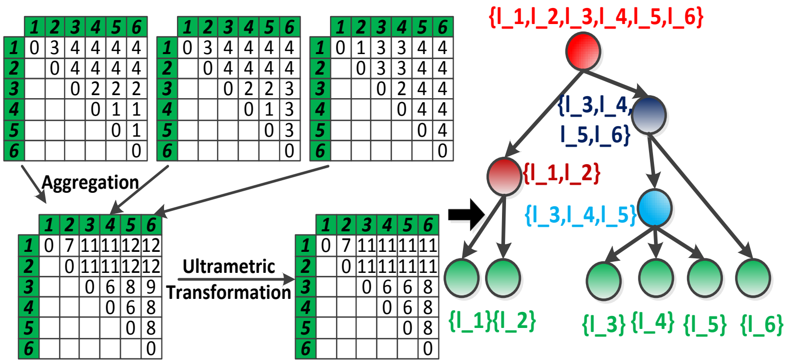


	1	2	3	4	5	6
1	0	3	4	4	4	4
2		0	4	4	4	4
3			0	2	2	2
4				0	1	1
5					0	1
6						0

Figure 3.6: A dendrogram descriptor and PMD derivation

7. Hierarchy aggregation: For a target user, the hierarchies of top-k matching clusters are ensembled to represent her hierarchical preferences. Although there are many popular descriptors (e.g., partition membership divergence (PMD), cophenetic divergence, cluster membership divergence (CMD)) for hierarchical structure comparison, the PMD was found to be better [PD84].

   The PMD gives the number of partitions in which the two objects in the hierarchy are not assigned together to a group. It preserves the order of hierarchical levels, has only fewer (2*n-4 values in an n*n matrix) nearest neighbor interchange affects, and the magnitude of changes is dependent on the number of hierarchical levels falling between the two levels undergoing the interchange. Figure 3.6 illustrates the PMD computation of a hierarchical structure where the items are grouped as (see the group ordering from bottom to top in Figure 3.6): {1}{2}{3}{4}{5}{6}; {1}{2}{3}{4,5,6}; {1}{2}{3,4,5,6}; {1,2 }{3,4,5,6}; {1,2,3,4,5,6}. Starting from the bottom of the hierarchy, all the nodes are isolated. As we go up, the nodes 4,5,6 get aggregated and the rest are still isolated. This level of aggregation results in the set {1}{2}{3}{4,5,6}. Similarly, in



	1	2	3	4	5	6
1	0	3	4	4	4	4
2		0	4	4	4	4
3			0	2	2	2
4				0	1	1
5					0	1
6						0

	1	2	3	4	5	6
1	0	3	4	4	4	4
2		0	4	4	4	4
3			0	2	2	3
4				0	1	3
5					0	3
6						0

	1	2	3	4	5	6
1	0	1	3	3	4	4
2		0	3	3	4	4
3			0	2	4	4
4				0	4	4
5					0	4
6						0

	1	2	3	4	5	6
1	0	7	11	11	12	12
2		0	11	11	12	12
3			0	6	8	9
4				0	6	8
5					0	8
6						0

	1	2	3	4	5	6
1	0	7	11	11	11	11
2		0	11	11	11	11
3			0	6	6	8
4				0	6	8
5					0	8
6						0

Figure 3.7: Hierarchy aggregation and ultrametric transformation

the next level, the node 3 also gets aggregated to the set {4,5,6}and the resulting aggregation becomes {1}{2}{3,4,5,6}, and so on.

The aggregation is repeated until all the nodes are aggregated into a single set (see Figure 3.6 for detail). After we get different level of aggregations, derivation of the PMD matrix can be achieved by counting the number of levels where two nodes are not assigned to same set. For instance, the node 3 and node 4 are on different set at the bottom level (where all nodes have their individual set) and at the first level (where the node 4 gets merged with nodes 5 and 6 to get {4,5,6}but the node 3 still exists with its own set), hence the PMD matrix has the value 2 for the nodes 3 and 4. The PMD matrix can be populated accordingly.

We do an element-wise aggregation of the PMDs to get a single PMD (see Figure 3.7). The goal of agglomerative hierarchical clustering is to ensure that the closest clusters get merged, however, the aggregated PMD does not ensure that the closest clusters get merged.

In order to ensure the closest clusters merge, we use the concept of ultrametric space which is more strict than triangle inequality. As the aggregated PMD may not satisfy ultrametric property, the hierarchical structure obtained from this metric may not be topologically correct. The correct topology ensures proper order of merging of closest clusters. In order to achieve this, we transform the aggregated PMD into an ultrametric form.

The ultrametric distance is an approximation of the distance matrix, which can be derived from the aggregated PMD. Any distance matrix $dist_{I\times J}$ is ultrametric *iff* following conditions hold:

(a) **non-negativity** ($a \neq b, dist(a,b){>}0$),

(b) **symmetry** ($dist(a,b) = dist(b,a)$), and

(c) **ultrametricity** ($dist(a,c) \leq \max(dist(a,b), dist(b,c))$

Intuitively, the hierarchical clustering merges closest clusters C_i and C_j if the following distance property is satisfied:

$$dist(C_i, C_j) \leq min(dist(C_i, C_k), dist(C_j, C_k)). \tag{3.28}$$

This also implies that:

$$\forall_{i,j,k}, min(dist(C_i, C_k), dist(C_j, C_k)) \leq dist(C_i \cup C_j, C_k). \tag{3.29}$$

This reducibility condition [HJ97] illustrates that the merge takes place between closest pairs and maintains the initial merge order. As long as the reducibility condition is satisfied, the updated dissimilarities satisfy the ultra-metric inequality [HJ97]:

$$dist(x_i, x_j) \leq max(dist(x_i, x_k), dist(x_j, x_k)), \forall_{x_i, x_j, x_k \in X}. \tag{3.30}$$

We use the transitive dissimilarity $T(P_{ij})$ of any path P_{ij} between vertices V_i and V_j which is defined as:

$$T(P_{ij}) = max(dist(i, k_1), ..., dist(k_{n-1}, k_n), dist(k_n, j)). \tag{3.31}$$

The minimal transitive dissimilarity is defined as the minimum of transitive dissimilarity among all paths between vertices V_i and V_j: $m_{ij} =$

$\min_{P_{ij}}(T(P_{ij}))$. The minimal transitive dissimilarity between any two vertices satisfy ultrametric inequality [ZLD10]. As the minimal transitive dissimilarity satisfies the ultrametric inequality: $m_{ij} \leq max(m_{ik}, m_{jk}), \forall_{i,j,k}$ [ZLHL13], we exploit the modified Floyd-Warshall algorithm [DHX+06] to find the new transitive dissimilarity matrix which is the closest approximation of the original matrix and is also ultrametric.

Figure 3.7 presents the PMD aggregation and ultrametric transformation process. Algorithm 1 shows the steps for the transformation of aggregated PMD matrix. After the ultrametric transformation, we can use any hierarchical clustering method to get the aggregated hierarchy from the transformed matrix.

Algorithm 1 FindTransitiveDissimilarityMatrix

```
Input G: the pair-wise distance matrix
Output H: minimum transitive dissimilarity matrix closure of G
Initialize H to G
for k=1 to N do
    for i=1 to N do
        for j =1 to N do
            $H_{i,j} = min(H_{i,j}, max(H_{i,k}, H_{j,k}))$
        end for
    end for
end for
return H
```

8. Recommendation generation: We traverse the aggregated hierarchical structure to generate a recommendation.

    (a) POI Recommendation: The ensembled hierarchy for a region is traversed to find the best match between a user profile, current context, and the items at each level. The set of items at each node of hierarchy represents items that match similar preferences of users. At each level of the hierarchy, we compute the preference score of the user on the available branching nodes, and traverse on the branch that has high preference score and has the best contextual match between item profile and current context. The process is repeated until we reach a leaf node. Any already recommended item is ignored in next item recommendation. As the users' preference score (see Eqn. 3.17) incorporates different contexts, the traversal of aggregated hierarchy ensures the best contextual match of users' preferences and the items in the hierarchy.

(b) POI sequence recommendation: The sequence generation is based on the previous POI recommended. For instance, given a user's current location, the first step is to identify the best matching region for the user. This is accomplished by finding the k-nearest regions to the user's current location. The trees of these regions are traversed using the current context and preference score of the user.

From the best matching leaf node (i.e. a set of locations that satisfy the context across the path/branch with maximum preference score and with the items' profiles matching the current context), the location with highest preference score (the contextual preference as defined in Eqn. 3.17) is added to the sequence. We skip the previously selected item during traversal, traverse the tree from the root node, and follow a branch (i.e. consider next high scored first level branch and so on). Using this approach, we generate k-sequences that match the current context.

A simple approach will be to perform DFS (depth-first search) traversal on the same hierarchy to generate remaining recommendations. The evaluation section presents this model as HiRecSI. As removing an item from the item pool can have an effect on the user clusters, an interesting approach would be to repeat the user clustering, hierarchy generation, hierarchy aggregation, and DFS traversal to get next recommended item. The evaluation section presents this model as HiRecSII.

3.5.3 Evaluation

This section presents the dataset, evaluation metrics, baseline and relevant models, and the experimental settings.

Dataset

We used the Weeplace[5] and Gowalla [LLAM13] dataset collected from two popular LBSNs - Gowalla and Weeplaces. These datasets are well defined and have all attributes relevant to our study, such as (i) the location category, (ii) geospatial coordinates, (iii) friendship information, and (iv) check-in time. The statistics of the dataset is shown in Table 3.7. The top five checked-in categories and the check-in counts are shown in Table 3.8.

Dataset	**Check-ins**	**Users**	**POI**	**Links**	**POI Categories**
Gowalla	36,001,959	319,063	2,844,076	337,545	629
Weeplace	7,658,368	15,799	971,309	59,970	96

Table 3.7: Statistics of the datasets.

The work or home related category “Home/Work/ Other:Corporate/Office” was popular from 6 am to 6 pm, with the highest check-ins (42,019) made at 1 pm. Similarly, the “bars” had the highest of 21,806 check-ins at 2 am and the lowest check-ins (15,209) at 5 am. Most of the check-ins were at 12 pm - 6 pm and were either in home or work related categories.

[5] http://www.yongliu.org/datasets/

Weeplace dataset		Gowalla dataset	
Categories	No. of Checkins	Categories	No. of Checkins
Home/Work/Other: Corporate/Office	437,730	Corporate Office	1,660,159
Food:Coffee Shop	267,572	Coffee Shop	988,999
Nightlife:Bar	248,563	Mall	872,873
Shops:Food & Drink: Grocery Supermarket	160,913	Grocery	820,326
Travel: Train Station	152,104	Gas & Automotive	806,916

Table 3.8: Check-in counts on top five categories of Weeplace and Gowalla dataset.

Evaluation Metrics

We evaluate the performance of POI recommendation using precision, recall, and F-Score metrics. The correctness of the POI sequence is evaluated using diversity, displacement, and NDCG metrics. The diversity metric [BS01] of a sequence measures the variety of category and is measured using their categorical similarity (i.e. Similarity =1 if two places are of same category and Similarity = 0 otherwise). The high value of diversity means the list of items is more diverse in category.

$$\text{Diversity } (c_1, c_2, ..., c_n) = \frac{\sum\limits_{i=1}^{n}\sum\limits_{j=i+1}^{n}(1 - \text{Similarity } (c_i, c_j))}{\frac{n}{2} * (n-1)}. \tag{3.32}$$

The displacement measures the distance (in Km) between the predicted sequence (seq_a) and actual sequence (seq_e):

$$\text{Displacement } (seq_a, seq_e) = \sum_{i=1}^{k} \mid \text{Distance}(seq_{a_i}, seq_{e_i}) \mid \tag{3.33}$$

A high displacement means the predicted list items are far from the actual ones. We evaluate the generated sequence using normalized discounted cumulative gain (NDCG) metric:

$$\begin{aligned} NDCG_N &= \frac{DCG_N}{IDCG_N}, \\ DCG_N &= \sum_{i=1}^{N} \frac{2^{rel_i} - 1}{\log_2(i+1)}, \\ IDCG_N &= \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{\log_2(i+1)}, \end{aligned} \tag{3.34}$$

where DCG$_N$ is the discounted cumulative gain from the relevance score of all items up to position N, rel_i is the relevance score of ith item in the generated sequence, IDCG$_N$ is the ideal discounted cumulative gain, and $\mid REL \mid$ is the list of relevant items in the dataset up to position N.

Evaluation Baselines

We evaluate the POI recommendation performance against the following baselines:

1. POI Popularity: It is a naive approach that uses the popularity of POIs. An area within a predefined radius is used to find the most popular POI (i.e. most visits in the locality) within it. The radius is dynamically updated by a predefined factor when no location is found in the area.
2. UCF: It is a user-based collaborative filtering model that relies on the user-item matrix and uses the cosine similarity to measure user-user similarity.
3. UCF+G [YYLL11]: It is an extension of UCF model that integrates geographical information into user-based CF in a linear interpolation fashion.
4. GeoMF [LZX$^+$14]: It is a state-of-art POI recommender that first incorporates the geographical information into matrix factorization by combining users' activity area vectors and POIs' influence area vectors into original users' latent factors and POIs' latent factors.

5. HSR [WTWL15]: It is a matrix factorization framework which explores the implicit hierarchical structures of users and items simultaneously for recommendation. It overcomes the gap between the importance of hierarchical structures and their unavailability.

6. Hierarchical Geographical Matrix Factorization model (HGMF) [ZXL$^+$17]: It is an extension of GeoMF [LZX$^+$14] and uses a two-dimensional normal distribution to represent the extent of POI influence over a geographical region. It then exploits matrix factorization on user content preference matrix, user spatial preference matrix, and POIs characteristic matrix jointly by modeling the implicit hierarchical structures, which is learned with an optimization process.

We use following baselines to evaluate the performance of sequence recommendation:

1. POI Popularity model as defined above.

2. Markov Chain-based approach: A first-order Markov Chain is used to generate the sequences. A Laplace smoothed state-transition and initial probability matrices are derived from the check-in data and are personalized for each user.

3. Hierarchical Geographical Matrix Factorization model (HGMF) [ZXL$^+$17].

4. Recurrent Neural Network (RNN): RNN-based sequence models are quite popular in the language domain (e.g., machine translation). In our case, it is a simple vanilla RNN-based model which uses the embedding of input sequence and generates output sequences. We also consider the Long-short term memory (LSTM) and Gated Recurrent Units (GRU).

5. Spatio-Temporal RNN (ST-RNN) from Liu et al. [LWWT16]: It incorporates local temporal contexts in each layer of RNN to model sequential

elements. It utilizes the recurrent structure to capture the periodical temporal contexts and employs time-specific and distance-specific transition matrices to characterize contextually dynamic location sequences.

Experimental Settings

We used a 5-fold cross-validation to measure the performance of the models. An Ubuntu 14.04.5 LTS, 32 GB RAM, a Quadcore Intel(R) Core(TM) i7-3820 CPU @ 3.60 GHz was used to evaluate the models. The same configuration with a Tesla K20c 6 GB GPU was used to evaluate the neural network-based models. For each user, the 10 most frequently checked-in places from the test set were taken as starting point, and 10 sequences per starting point was generated. The average metrics on the generated sequences were observed. The POI-Popularity used distance threshold of 2 Km.

The RNN model used 5 layers and 256 nodes. The input sequence length was set to 25, the data was fed in batches of size 50, embedding vectors were of size 384, and the experiment was repeated for 100 epochs. The learning rate was set to 0.002, and the gradients were clipped at 5 to prevent overfitting. For ST-RNN, the parameters were estimated using Back Propagation Through Time (BPTT) [RHW86]. In HGMF, following the original work, the number of sub-categories in the second layer of user spatial implicit hierarchical structure was taken as {50, 100, 200, 400, 800}, the number of user latent sub-categories in the second layer was taken as {50, 100, 200, 400, 800}, the POI latent subcategories was taken as {100, 200, 400, 800, 1000}, and the dimension of the latent factors was set to 100.

The parameters used in our models can be estimated by using the relevant fraction of check-ins from our training dataset (e.g., for social impact, we check the fraction of check-ins that are common among friends). Our observation found that most of these relevant fraction of check-ins were ∼0.25, or ∼0.5, so we defined three different values $\{0.25, 0.5, 0.75\}$ and observed the performance of our models on these values of the relevant parameters. For Eqn. 3.14, we observed the impact of three different values $\alpha \in \{0.25, 0.5, 0.75\}$ and found that $\alpha = 0.25$ gave the better result.

For Eqn. 3.15, we repeated with the same set of values and found better result with $\psi_1 = 0.5$. For the preference score defined in Equation 3.17, we also used the three set of values $\{0.25, 0.5, 0.75\}$ for θ and β. Our models performed better when $\beta = 0.25$ and $\theta = 0.25$. Although, these parameters might vary on the nature of dataset, our observation on all the experimental datasets found that the temporal factor should be weighted more than the categorical factor.

Experimental Results and Discussion

The precision, recall, and F-score of different models are presented in Table 3.9. The precision@N and recall@N performance of different models are presented in Table 3.10 and Table 3.11. The average diversity of different models in Weeplace and Gowalla dataset is presented in Table 3.12 and the displacement on Weeplace and Gowalla dataset is presented in Table 3.13. The displacement trend with increasing sequence length is shown in Figure 3.8 and Figure 3.9 and the diversity trend is shown in Figure 3.10 and Figure 3.11. The NDCG performance of different models is presented in Table 3.14.

Discussion

The popularity-based model performed worst among all the models. It generated almost similar sequences for all the users and was not relevant to personalized preferences. This might be due to the ignorance of personalized user preferences. The diversity measure was also quite low, which means the POIs in the generated sequences included few categories. The high displacement

	Weeplace dataset			Gowalla dataset		
Models	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**
Popularity	0.0098	0.0038	0.0054	0.0101	0.0041	0.0058
UCF	0.0249	0.0284	0.0265	0.0172	0.0295	0.0217
UCF+G	0.0302	0.0303	0.0302	0.0191	0.0322	0.0239
GeoMF[LZX+14]	0.0609	0.0545	0.0575	0.0450	0.0663	0.0536
HSR [WTWL15]	0.0441	0.0306	0.0361	0.0273	0.0402	0.0325
HGMF [ZXL+17]	0.0672	0.0560	0.0611	0.0526	0.0644	0.0579
HiRecSI	0.0725	0.0591	0.06512	0.0577	0.0653	0.0612
HiRecSII	0.0738	0.0610	**0.0668***	0.0621	0.0659	**0.0639***

Table 3.9: Precision, Recall, and F-Score of different models (* means statistically significant at 95% confidence level)

	Weeplace dataset				Gowalla dataset			
Models	**@5**	**@10**	**@15**	**@20**	**@5**	**@10**	**@15**	**@20**
Popularity	0.0106	0.0101	0.0094	0.0091	0.0108	0.0103	0.0098	0.0094
UCF	0.0281	0.0253	0.0237	0.0227	0.0201	0.0187	0.0152	0.0148
UCF+G	0.0328	0.0305	0.0293	0.0283	0.0252	0.0194	0.0162	0.0157
GeoMF[LZX+14]	0.0648	0.0615	0.0591	0.0585	0.0557	0.0482	0.0392	0.0371
HSR [WTWL15]	0.0452	0.04481	0.0456	0.0410	0.0361	0.0287	0.0226	0.0221
HGMF [ZXL+17]	0.0724	0.0683	0.0655	0.0627	0.0626	0.0574	0.0457	0.0450
HiRecSI	0.0774	0.0743	0.0710	0.0673	0.0638	0.0620	0.0541	0.0510
HiRecSII	0.0781	0.0754	0.0727	0.0692	0.0661	0.0640	0.0613	0.0572

Table 3.10: Precision@N Performance of different models

	Weeplace dataset				Gowalla dataset			
Models	**@5**	**@10**	**@15**	**@20**	**@5**	**@10**	**@15**	**@20**
Popularity	0.0014	0.0018	0.0020	0.0101	0.0016	0.0018	0.0022	0.0107
UCF	0.0214	0.0257	0.0324	0.0341	0.0221	0.0253	0.0344	0.0362
UCF+G	0.0235	0.0278	0.0341	0.0360	0.0251	0.0311	0.0357	0.0370
GeoMF[LZX+14]	0.0453	0.0526	0.0579	0.0625	0.0521	0.0611	0.0750	0.0772
HSR [WTWL15]	0.0271	0.0301	0.0326	0.0326	0.0311	0.0357	0.0452	0.0491
HGMF [ZXL+17]	0.0411	0.0566	0.0612	0.0652	0.0574	0.0610	0.0682	0.0711
HiRecSI	0.0473	0.0583	0.0636	0.0672	0.0568	0.0626	0.0693	0.0725
HiRecSII	0.0477	0.0590	0.0677	0.0696	0.0570	0.0633	0.0700	0.0735

Table 3.11: Recall@N Performance of different models

Models	**Diversity in Weeplace**	**Diversity in Gowalla**
Popularity	1.2000	3.2000
Markov	2.5000	3.6000
HGMF	6.9110	7.6600
RNN	7.0010	7.8301
LSTM	7.1100	7.9026
GRU	7.2250	7.9400
ST-RNN	7.3301	8.1826
HiRecSI	7.2180	8.0476
HiRecSII	**7.4912**	**8.4500**

Table 3.12: Diversity in Weeplace and Gowalla dataset on sequence length of 25

Models	**Displacement in Weeplace**	**Displacement in Gowalla**
Popularity	25.3078	25.2287
Markov	16.7213	13.2211
HGMF	9.1916	9.5931
RNN	8.6103	9.0715
LSTM	8.3251	8.7934
GRU	8.1051	8.4820
ST-RNN	7.9600	7.8620
HiRecSI	7.7001	7.7766
HiRecSII	**7.3701**	**7.7100**

Table 3.13: Displacement (Km) in Weeplace and Gowalla dataset on sequence length of 25

	Weeplace dataset			**Gowalla dataset**		
Models	**NDCG**$_{10}$	**NDCG**$_{20}$	**NDCG**$_{30}$	**NDCG**$_{10}$	**NDCG**$_{20}$	**NDCG**$_{30}$
Popularity	0.2867	0.2892	0.2895	0.2885	0.2901	0.2975
Markov	0.2979	0.3009	0.3079	0.2989	0.3103	0.3119
HGMF	0.4210	0.4258	0.4372	0.4251	0.4278	0.4372
RNN	0.4536	0.4696	0.4783	0.4566	0.4702	0.4785
LSTM	0.4844	0.4926	0.4983	0.4661	0.4882	0.4892
GRU	0.5262	0.5337	0.5381	0.5427	0.5551	0.5581
ST-RNN	0.5633	0.5639	0.5679	0.5683	0.5701	0.5759
HiRecSI	0.5565	0.5600	0.5601	0.5625	0.5685	0.5715
HiRecSII	**0.5771**	**0.5799**	**0.5803**	**0.5791**	**0.5799**	**0.5873**

Table 3.14: NDCG$_N$ of different models on Weeplace dataset

metrics indicate that the predicted POIs were far from the actual ones. The NDCG metric was also least for popularity-based model.

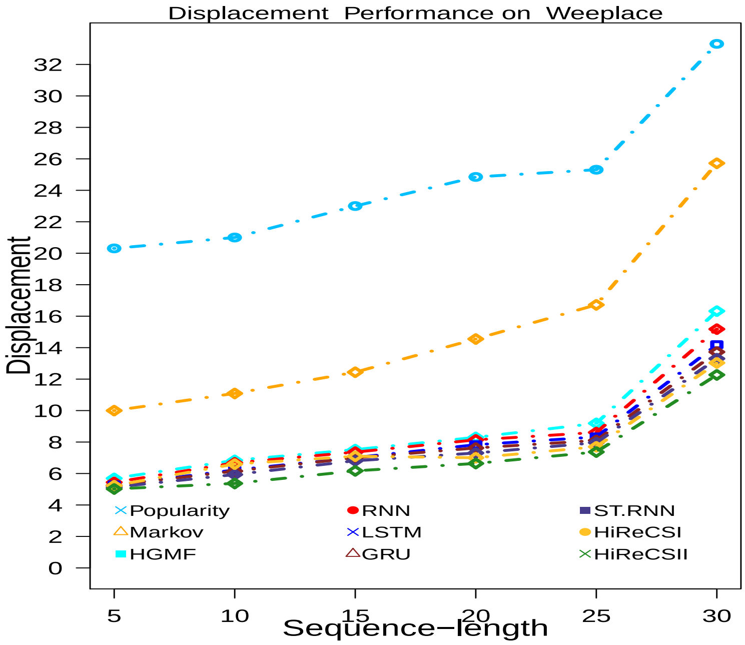


Figure 3.8: Displacement trend in Weeplace

The UCF and UCF-G models were better than Popularity model but were outperformed by the contextual models GeoMF and HSR. The HGMF outperformed HSR and has slightly lower performance than HiReCSI. HiReCSII outperformed all the other models in terms of precision, recall and F-Score metrics.

For the generated sequences, Popularity model performed least. The first-order Markov model relied on one previous check-in data to determine next location and hence was not able to fully model the check-in sequence generation process. However, its diversity, and displacement metrics were better than popularity-based model which is due to the personalization implied from separate initial-probability and state-transition tables for each user. The HGMF modeled the hierarchical relation between user and item latent factors and outperformed Markov model on both datasets and on all the evaluation metrics. However, its performance was lower than RNN because sequence modeling and

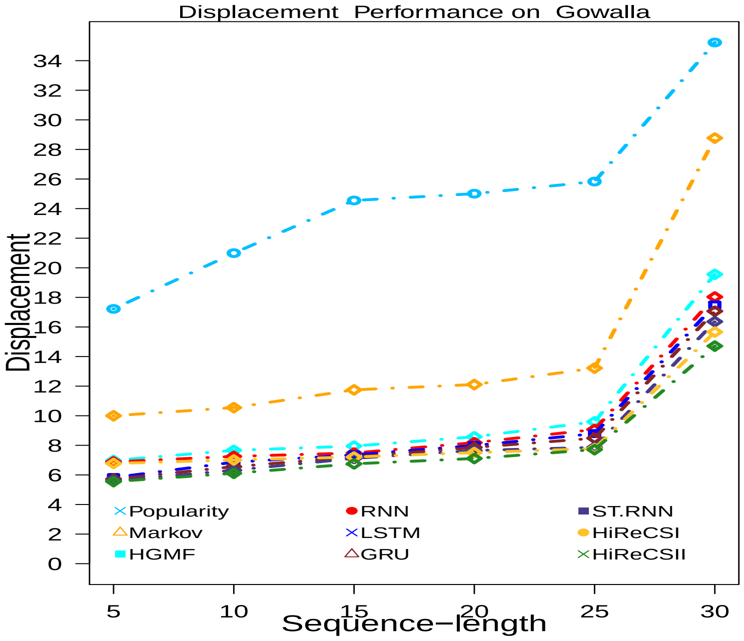


Figure 3.9: Displacement trend in Gowalla

locality preference aggregation was not handled efficiently in this model. The regular RNN models handled the sequence modeling better than HGMF but did not incorporate the spatial and temporal contexts. The LSTM and GRU performed slightly better than RNN. The ST-RNN model incorporated the sequence along with the spatiotemporal contexts, and was better than regular RNN model. The performance of ST-RNN was in par with HiRecSI in terms of diversity and NDCG metrics. However, HiRecSI was better in terms of displacement.

The HiRecSII model outperformed all the other models. Its performance was slightly better than HiRecSI and ST-RNN in both datasets. The performance of HiReCSI and HiReCSII was in par on displacement on Gowalla dataset. The HiRecSII regenerated the hierarchy once an item is selected for an output sequence. This ensures better modeling of the similarities between the remaining items, and hence results in better preference hierarchy. On the other hand, HiRecsI generates the hierarchy only once, and uses it to generate the whole list.

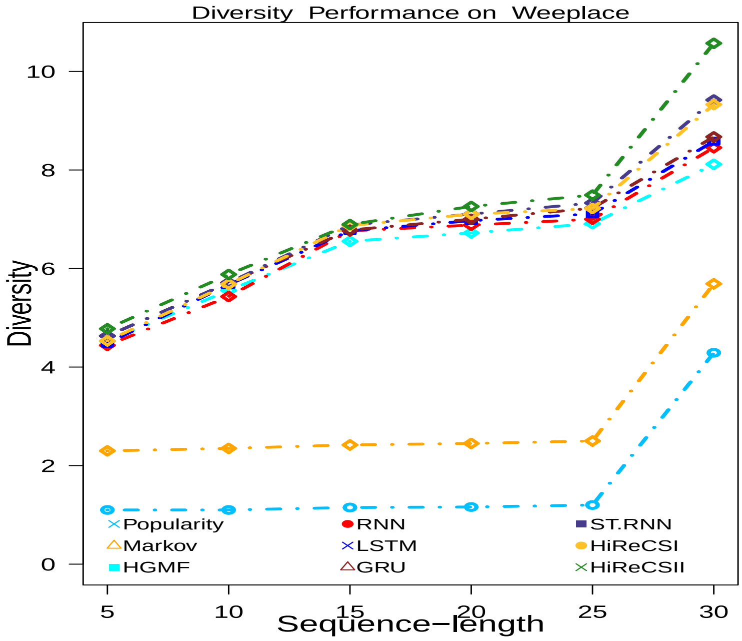


Figure 3.10: Diversity trend in Weeplace

Impact of number of regions and clusters

To better incorporate the spatial distribution of coherent POIs, we analyzed the impact of area of regions on the length of sequences on the dataset. The grid regions on every 5, 10, 15, and 20 Km distances were analyzed. With 5 Km, the average sequence length was 10 (i.e. 10 different check-ins within the region) and it was 18, 25, 30 for 10, 15, and 20 Km respectively. So, we selected the grids with 20 Km overlapped by 1 Km to have reasonable sequence length. As ensembling of many clusters implies preference aggregation from many users which might not result in best preference match, we used an aggregation of top 5 clusters that matched to a target user.

Case Studies on POI sequence generation

We provide a case study on sequences generated by popularity-based model and HiRecS on Gowalla dataset[6]. We selected sequences of length 5 for two different users 'thadd-fiala', 'boon-yap' (known as u_1, u_2 now onwards) with most check-

[6] a similar trend was observed on Weeplace dataset as well

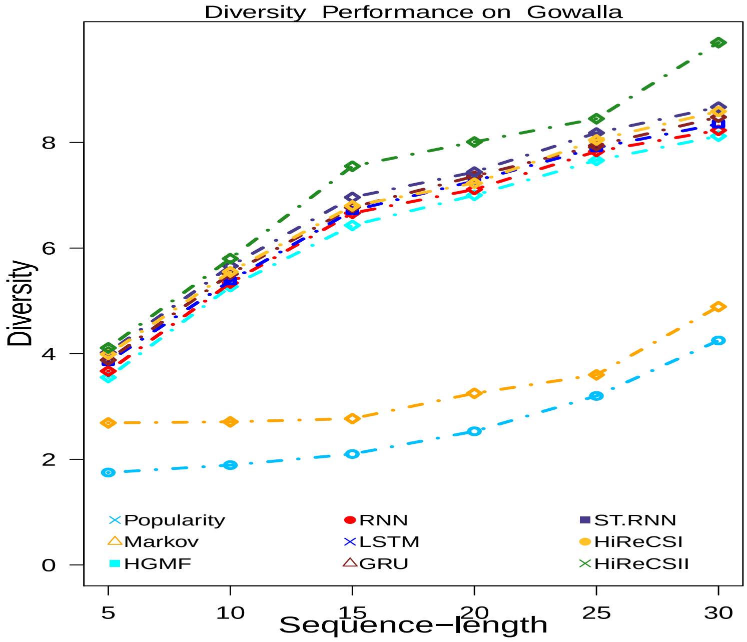


Figure 3.11: Diversity trend in Gowalla

ins and analyze the relevance of the sequences for them. For u_1, a sequence of length 5 from popularity-based model was {'sycamore-place-lofts-cincinnati', 'pg-gardens-cincinnati', 'lytle-park-cincinnati', 'piatt-park-cincinnati', 'sycamore-place-at-st-xavier-park-apartments-cincin'}, and their respective categories were {'Home/Work/Other:Home', 'Parks & Outdoors: Plaza / Square', 'Parks & Outdoors:Park', 'Parks & Outdoors: Plaza / Square', 'Home/Work/Other: Home'}. Similarly for user u_2 a length 5 sequence was {'starbucks-boston', 'mbta-south-station-boston', 'boston-common-boston', 'dunkin-donuts-boston', 'mbta-park-street-station-boston'} and their respective categories were {'Food: Coffee Shop', 'Travel: Train Station', 'Parks & Outdoors:Park', 'Food: Coffee Shop', 'Travel:Train Station'}. Most of the places recommended were the popular ones and the generated sequences had less diversity. For both users, there were three different categories in the generated sequences. With the increasing sequence length, the diversity showed some increasing trend (see Figure 3.10 and Figure 3.11) but this was lower in both datasets.

With HiRecSII, a sequence generated for user u_1 was {'sycamore-place-lofts-cincinnati', 'pg-gardens-cincinnati', 'piatt-park-cincinnati', 'lytle-park-cincinnati', 'lpk-cincinnati'} and their categories were {'Home/ Work/ Other: Home', 'Parks & Outdoors: Plaza/Square', 'Parks & Outdoors: Plaza/Square', 'Parks & Outdoors: Park', 'Home/Work/ Other: Corporate/ Office'}. For user u_2 a sequence was {'starbucks-boston', 'mbta-park-street-station-boston', 'boston-common-boston', 'digitas-boston-boston', 'hubspot-cambridge'} and their categories were {'Food:Coffee Shop', 'Travel:Train Station', 'Parks & Outdoors: Park', 'Nightlife: Speakeasy / Secret Spot', 'Home/Work/Other: Corporate/ Office'}. We can observe that for both users, there are at least four different categories in the sequence and the recommendations were more contextual. With the increasing sequence length, the diversity showed some increasing trend (see Figure 3.10 and 3.11) which was best with HiRecSII.

With the popularity-based model, the average displacement of the above sequence was 19.36 Km for user u_1 and it was 20.03 Km for user u_2. With HiRecSII, the average displacement of the above sequence was 5.02 Km for user u_1 and it was 5.61 Km for user u_2. This shows that HiRecSII addresses the distance constraint better. With the increasing sequence length, the displacement trend increased for both models and followed the trend as shown in Figure 3.8 and 3.9.

3.5.4 Conclusion and Future Work

We modeled user activity and location influence to generate context-aware recommendation. Evaluation of the proposed model on two real-world datasets demonstrated the efficiency of proposed model. We also formulated the contextual and locality-based user preferences in terms of hierarchy and presented a hierarchy aggregation technique to generate POI sequence recommendation. We defined user preferences using different contexts (e.g., social, temporal, categorical, and spatial) and generated POI sequences to match both the locality preferences and user preferences.

We extensively evaluated the performance of the proposed models using F-score, diversity, displacement, and NDCG metrics on two real-world datasets. We demonstrated the significant performance gain using our model (of 0.006 - 5.91 on diversity, 0.0349 - 17.51 on displacement, and 0.114 - 0.289 on NDCG metrics) when compared to several baseline models and relevant studies. There are many interesting directions to explore as future studies. We would like to incorporate the textual (e.g., tags, tips, and review text) and also visual information (e.g., the image of places) to define the preferences of users and the popularity of places. We would like to exploit the knowledge discovery from the aggregated hierarchy and like to extend the model for group recommendation.

CHAPTER 4

REVIEW-AWARE EXPLANATION OF RECOMMENDATION

The Location-Based Social Networks (LBSN) (e.g., Facebook, etc.) have many attributes (e.g., ratings, reviews, etc.) that play a crucial role for the Point-of-Interest (POI) recommendations. Unlike ratings, the reviews can help users to elaborate their consumption experience in terms of relevant factors of interest (aspects). Though some of the existing systems have exploited user reviews, most of them are less transparent and non-interpretable (as they conceal the reason behind recommendation). These reasons have motivated us towards explainable and interpretable recommendation. To the best of our knowledge, only few of the researchers have exploited user reviews to incorporate the sentiment and opinions on different aspects for personalized and explainable POI recommendation.

This paper proposes a model termed as ReEL (Review aware Explanation of Location Recommendation) which models the review-aspect correlation by exploiting deep neural network, formulates user-aspect bipartite relation as a bipartite graph, and models the explainable recommendation by using dense subgraph extraction and ranking-based techniques. The major contributions of this paper are: (i) it models users and POIs using the aspects posted on user reviews and provisions incorporation of multiple contexts (e.g., categorical, spatial, etc.) in POI recommendation, (ii) it formulates preference of users' on aspects as a bipartite relation, represents it as a location-aspect bipartite graph, and models the explainable recommendation with the notion of ordered dense subgraph extraction using bipartite cores, shingles, and ranking techniques, and (iii) it extensively evaluates the proposed models with three real-world datasets and demonstrates an improvement of 5.8% to 29.5% on F-score metric over relevant studies.

4.1 Introduction

Most of the existing e-commerce systems (e.g., Amazon.com, etc.) have been facilitating users to share their consumption experience via ratings and reviews. The LBSNs have also been a useful platform to share consumption experiences on different factors of interest (e.g., price, service, accessibility, product quality, etc.). For instance, the review text *"The breakfast was awesome but the front-desk service was really bad"* implies a positive experience of the reviewer towards "breakfast" and opposite for "front-desk". The words "breakfast" and "front-desk" are known as **aspect terms** and their equivalent categories "Food" and "Service" are known as **aspects**. Such experiences from a real customer have been crucial in the purchase decision for potential customers, and product improvement for manufacturers.

Despite the usefulness, reading time and uniform interpretability of reviews have been a major concern. It would have been easier if one can summarize and explain the opinions on key aspects, for instance, (i) *place A* has a *good* rating for *food*, (ii) *place B* is *renowned* for *cleanliness*, etc. Though a dedicated community has been focusing on the extraction of such aspects and opinions [WPDX17, WHZ$^+$16, CZZ$^+$17], the recommendation domain can also use such aspect-based summarization to enhance and explain the generated recommendation.

The exploitation of different factors of LBSN for an efficient recommendation has been quite popular in the last decade [YCM$^+$13, ZYHW16]. Most of the studies have focused on non-text attributes, such as categorical, temporal, spatial, and social [BL16, XLLZ18, BL17, XNL$^+$17, BWLC16] but have been less transparent and less interpretable (i.e. the factors used for recommendation are hidden from end users). Contrary to that, some of the studies [VSR09, SNM08, TM12, GJG14, MLS16, GGJ11, ZCYZ15, MNL$^+$16] have already claimed the user persuasiveness due to explainability in real-world systems. The similarity-based approaches [HKR00, BM05] have proposed user-based neighbor style (e.g., *"users with similar interest have purchased the following items..."*) explanations. The item-based neighbor style (e.g., *"items similar to you viewed or purchased in the past..."*), influence style (how the users' input have influenced the generation of recommendation), and keyword-style (items that have similar description content to purchase history) can be other variants of explanations.

To the best of our knowledge, only few studies have focused on review-aware explainable recommendation. There are many factors that make this problem challenging and interesting. The aspect extraction from ambiguous and noisy text, organizing the numerous aspect terms into relevant categories (e.g., food, service, etc.), and personalization of recommendation are some of the main challenges. The aspect-based personalized explanation is challenging as it needs to handle the sentiments of each aspects, and also the individual user preferences and item features to get relevant explanation.

The ease of adaptation of arbitrary continuous and categorical attributes in a scalable manner makes the Convolutional Neural Networks (CNN) a good candidate for classification problems (e.g., [Kim14, CWB+11]). This also makes them ideal for a supervised review-aspect classification problem. We formulate the problem of review and aspect correlation using CNNs. This simplifies the process of mapping the user sentiments to the $\langle$POI, aspect$\rangle$ tuples and modeling the users' aspect preferences as the aspect-POI bipartite relation. We represent such a bipartite relation using a bipartite graph, extract users' ordered aspect preferences using dense subgraph extraction and ranking-based methods, and generate an explainable POI recommendation. The core contributions of this chapter are: (i) it models users and POIs using the aspects extracted from reviews and different contexts (e.g., categorical, spatial, etc.),

(ii) it formulates the user preferences as an ordered aspect-POI bipartite relation, represents it as a bipartite graph, and proposes bipartite core, shingles, and ranking-based methods to generate personalized and explainable POI recommendation, and (iii) it evaluates the proposed model using three real-world datasets. As an important by-product, our model can implicitly identify the user communities and categorize them by their preferred aspects. It can also identify the implicit POI groups that are known for a set of aspects.

4.2 Related Research

The problem of aspect extraction from review text has been quite popular [LWZ12, ZLXJ11, CBdG$^+$17] for various problems (e.g., rating prediction [ML13], aspect-sentiment summarization [TM08, ME11, JO11], recommendation [ZCZ15, MMO16], etc.). To the best of our knowledge, exploitation of aspects for explainable POI recommendation has been less explored. We present the relevant studies in following two categories:

4.2.1 Aspect-based approaches

Yang et al. [YZYW13] exploited sentiment lexicon (e.g., SentiWordNet)-based approach and defined user preferences based on tips, check-ins, and social relations but did not fully exploit user preferences at aspect level. Wang et al. [WZN$^+$15] exploited multi-modal (i.e. text, image, etc.) topics-based POI semantic similarity but ignored aspect level preference modeling and recommendation explanation. Covington et al. [CAS16] exploited different factors, such as users' activity history, demographics, etc., but did not incorporate opinions from user comments and also did not focus on recommendation for each aspect. Guo et al. [GSZ$^+$17] represented users, POIs, aspects, and geo-social relations with a graph and ranked the nodes to define the POI recommendations. Some of the studies [DS17] used the features extracted from user reviews to build user and item profiles and generated the recommendation. Zhang et al. [ZCZ15] used the aspect opinions, social, and geographical attributes to generate the recommendation. Chen et al. [CC15] used aspect-based user preferences in their recommendation. Recently, Zheng et al. [ZNY17] adapted [CWB^{+}11] to exploit user reviews and mapped user and item feature vectors into same space to estimate user-item rating. Our model has following advantages than [ZNY17]: (i) it uses sentiment polarity of reviews at sentence level rather than the whole review text, (ii) it learns to classify each review sentence into aspects and models users and places using these aspects and embedding of additional contexts (e.g., POI category, check-in time, etc.), and (iii) it efficiently exploits a bipartite core extraction, shingles extraction, and ranking-based methods to extract densely connected aspects and relevant POIs for an explainable recommendation.

4.2.2 Explanation-based approaches

Chen et al. [COX16] personalized ranking based tensor factorization model and used phrase-level sentiment analysis across multiple categories. They extracted aspect-sentiment pairs from review text and used Bayesian Personalized Ranking [RFGST09] to rank the features from user reviews. Finally, feature wise preference of a user was derived using the user-item-feature cube and rank of the feature obtained earlier. Zhang et al. [ZLZ^{+}14] used matrix factorization to estimate the missing values and a recommendation was made by matching the most favorite features of a user and properties of items. They used simple text templates to generate a feature-based explanation of positive and negative recommendations. However, incorporation of additional features (e.g., POI category) was not explored. Lawlor et al. [LMRS15] exploited sentiment-based approach to explain why a place might(not) be interesting to a user. For each aspect, they compared the recommended place to the alternatives and provided explanation (e.g., better (worse) than 90% (20%) of alternatives for *room quality* (*price*), etc.). However, they relied on frequency of aspects of POIs and users to get such relation and incorporation of additional features remained unexplored. He et al. [HCKC15] exploited tri-partite modeling of user-item-aspect tuples and used graph-based ranking to find the most relevant aspects of a user that match with relevant aspects of places. The common relevant aspects were used in the explanation. Li et al. [CW17] proposed an explanation interface to explain the tradeoff properties within a set of recommendations, in terms of their static specifications and feature sentiments. However, their interface requires users to explicitly provide their preference on different aspects.

We have found that only few of the existing studies have fused few additional attributes (e.g., social), whereas most of them had no provision for them. Most of the studies were tightly coupled to aspects and their sentiments, and analyzed influence of all aspects together. The influence of aspects among each other can have adverse impact on recommendation quality, for e.g., a place that is good in “Price” aspect might be opposite in “Service” aspect. A user who just cares about “Price” aspect might ignore some “Service” related problems in that place. So we need to minimize the influence of aspects among each other. This is crucial for aspect-based recommendation systems, and to the best of our knowledge, this direction is less explored and is still a viable research problem. We attempt to fill this gap by exploiting bipartite graph and dense subgraph extraction techniques. For a user, the most dense subgraph represents the set of most preferred aspects and places popular for those aspects. The dense subgraph extraction is followed by disconnecting the edges within the dense subgraph which ensures less interference from the aspects already discovered in previous dense subgraphs. This claim is also supported by our evaluation where one of our model ReEL-Core performs better than our another model ReEL-Rank (see Sec. 4.4.1, Sec. 4.4.3, and Sec. 4.5 for details on the performance of these models).

4.3 Methodology

The block diagram of proposed model is shown in Figure 4.1 and the high-level overview is shown in Figure 4.2. The core components of the proposed system are as follows:

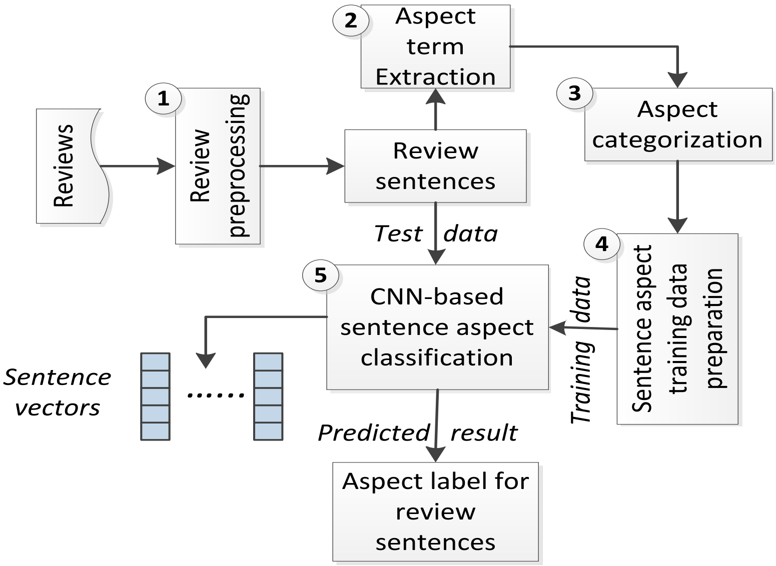


Figure 4.1: Block diagram of review classification module

4.3.1 Components of proposed model

In this section, we describe the individual components of the proposed model.

1. Review preprocessing: The review texts are splitted into individual sentences and the stop words are removed.

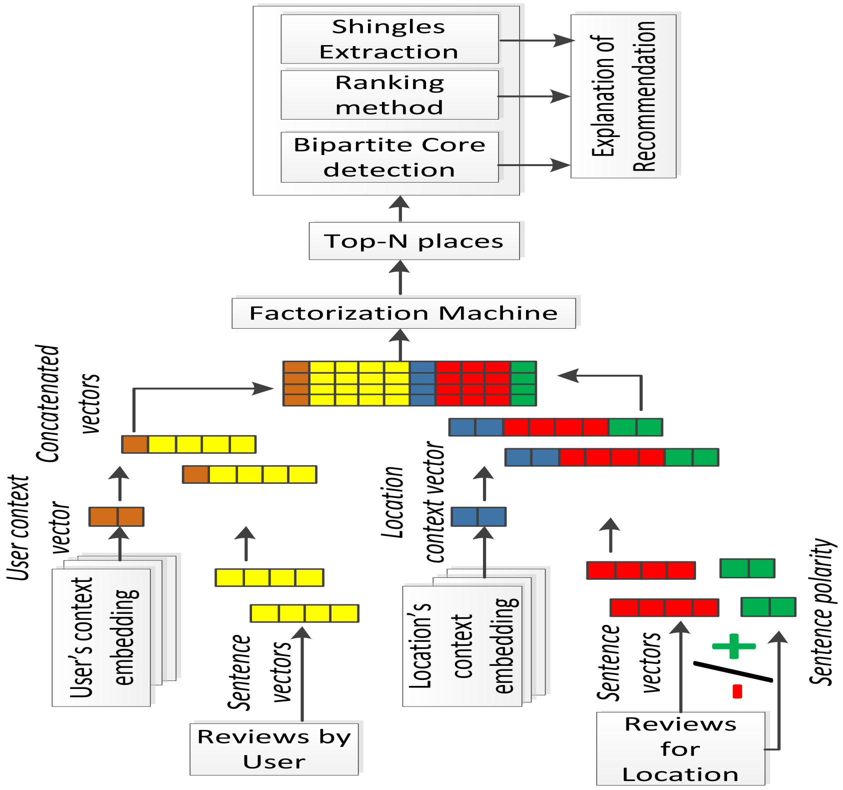


Figure 4.2: Overview of recommendation module

2. Aspect term extraction: The pre-processed review sentences are fed to the aspect extraction module to extract aspect terms. A simple two-step process is applied. First, we filter out nouns and noun phrases using some experimentally set frequency threshold. Most of the reviews focus on a set of topics, hence this approach can capture such topics [ME10]. Second, we use a rule-based approach [ZL14] that adopts the dependency parsing [MSB$^+$14] to capture the aspect terms missed in the previous step.

3. Aspect-categorization: As there can be numerous aspect terms, we narrow down them to few well-known aspects (see Table 4.1) for easy computation. The aspect terms and their synsets from WordNet [Fel98] are used to assign the best matching aspect. We select top 3 synsets to handle ambiguity of aspect terms and to capture the relevant aspect.

4. Sentence-aspect training data preparation: As the aspect extraction and labeling is not the core focus of this paper, we rely on supervised sentence-aspect classification concept. The review text (after aspect term extraction) is labeled by the aspect that has closest match to its aspect terms. The distance between aspect terms and the aspects (and their synonyms) from the WordNet [Fel98] are used to assign the closest possible label. As we assign top 3 matching synsets, a single aspect term can have three matching aspects. The sentences with multiple aspect terms get multiple label. This labeled data is used to train the CNN-based sentence-aspect classifier. The performance of this module is defined in the evaluation section (see Sec. 4.5).

5. CNN-based sentence-aspect classifier: The review-aspect correlation module is a multi-class classifier (see Figure 4.1) that classifies a review sentence into relevant aspects. Inspired from [Kim14], we use a CNN-based classifier to label each review sentence. The network consists of a convolution, an activation function, a max-pooling, a dense layer, and a softmax layer (see [Kim14] for detail). The input to this classifier is word embedding of review sentences. We use Word2Vec [MSC$^+$13] to map every word to a uniform size vector in a latent feature space. The outcome of the classifier is a bipartite relation between review and the aspects.

   For every user, the classifier gives a set of sentence feature vectors (later known as *user feature vectors* in this chapter) that are embedding of her preferred aspects. Similarly, for every POI, the sentence feature vectors (later known as *POI feature vectors* in this chapter) are embeddings of the aspects specified in its reviews. As every user tends to mention some opinion on preferred aspects in her reviews and every place is mentioned about the aspects it was reviewed for, such vectors incorporate the aspects relevant to users and POIs. As a POI can be positively or negatively reviewed for an aspect, we extract the sentiments of each review sentence by using the trigrams around the aspect terms. The embeddings of the sentiment term [MSC+13] is concatenated to the POI feature vector. As each POI can get multiple reviews on same aspect, the POI feature vector is normalized on feature vectors of each aspect. This review-aspect bipartite relation is then used to define the POI-aspect tuples and user-aspect tuples. Such a bipartite relation can be exploited to model user preferences via ordered aspect-POI relation using bipartite graph and dense subgraphs of such graph (see Sec. 4.4.1, and Sec. 4.4.3 for details). The POI-aspect pair is supplemented with the aggregated sentiment extracted from all the review sentences.

6. Recommendation generation: This variant of proposed model is termed as **D**eep **A**spect-based **P**OI recommender (DAP). Besides the review text, we also incorporate additional context (e.g., categorical, spatial, etc.) into the feature vector of the POIs obtained from the classifier.

   We formulate the recommendation problem as a matrix, whose rows represent a user, POI, and elements of different contexts. For each row, the check-in flag of a user to a POI is treated as the target. For instance, if a user u_i has feature vector as $\langle ue_1, ue_2, ..., ue_m \rangle$, a place l_j has its sentiment concatenated feature vector as $\langle le_1, le_2, ..., le_n \rangle$, and the user u_i has visited the place l_j, then a row in the design matrix is obtained simply by concatenating the user feature vector, POI feature vector, and context vectors, and is defined as: $\overrightarrow{u_i, l_j, f_k} = \langle u_ie_1, u_ie_2, ..., u_ie_m$ $, l_je_1, l_je_2, ..., l_je_n,\ f_ke_1, f_ke_2, ...f_ke_o, 1 \rangle$, where u_ie_a, l_je_a, and f_ke_a are the a^{th} item (a real-valued number) of the feature vector of user (u_i), place (l_j), and context (f_k). The last element *1* represents the check-in flag for the user-place-context tuple in the training data and represents the score to be estimated for the test data.

For a user u, the context vector is concatenation of temporal $\langle v_{t_1}, v_{t_2}, v_{t_3}\rangle$, spatial $\langle v_{dist_1}, v_{dist_2}, v_{dist_3}\rangle$, categorical $\langle v_{cat_1}, v_{cat_2}, ..., v_{cat_k}\rangle$, and social $\langle v_{soc}\rangle$ vectors. The term v_{cat_1} is the multiplication of embedding vector of category cat_1 and the factor $r_{cat_1} = \frac{\sum_{l.cat=cat_1} V_u(l)}{\sum_{l' \in u_L} V_u(l)}$ (i.e. the ratio of total check-ins made to places with category cat_1 to that of all check-ins). $v_{dist_1} = \frac{\sum_{dist(l) \leq \epsilon_1} V_u(l)}{\sum_{l' \in u_L} V_u(l')}$ is ratio of total check-ins on places within a threshold distance ϵ_1 (from users' home, work place or most frequently checked-in place) to that of all check-ins (we consider $\epsilon_1 \leq 1, 1<\epsilon_2 \leq 5, \epsilon_3>5$ as three distance thresholds (in K.M.)). $v_{soc} = \frac{\sum_{l \in u_{f_L}} V_u(l)}{\sum_{l' \in u_L} V_u(l')}$ is the ratio of total check-ins made on places visited due to social influence to that of all check-ins. v_{t_1} is the ratio of total check-ins made in time t_1 (we use three values for time - morning, afternoon, and others (night and evening)). The POI context vector consists of category, time, and distance vectors.

A factorization machine [Ren12] is exploited to estimate the value of the check-in flag for every user-place-context tuple. As the factorization machine has the ability to deal with additional features, a user-place pair can have multiple rows but just one row for each user-place-context tuple. So, the prediction is already personalized for the user-place-context tuple. The top-N scorers from factorization machine are further filtered out using the preferred aspects of user (determined by the frequency of aspects mentioned on her reviews) and are recommended to the users. The high-level overview of the recommendation module is illustrated in Figure 4.2.

7. Explanation of recommendation: After getting the place-aspect bipartite relation from CNN-based classifier, we represent the user-aspect preference as a bipartite graph and generate the recommendation explanation by extracting the most dense subgraphs from this bipartite graph.

Aspects	Example
Price	cheap, deals, coupons, cost
Food	food quality, food variety, free breakfast
Service	serving time, friendly staffs
Amenities	comfort, laundry, security, free parking, free WiFi
Accessibility	near, disability access, information on web
Others	security, pet friendly

Table 4.1: Aspects

We propose three different methods- a bipartite core extraction, shingles extraction, and ranking-based methods for explanation generation (see Sec. 4.4 for detail).

4.3.2 Factorization Machine

The Factorization Machine [Ren12] formulates the prediction problem as a design matrix $X \in \mathbb{R}^{n \times p}$. The i^{th} row $\vec{x}_i \in \mathbb{R}^p$ of the design matrix defines a case with p real-valued variables. The main goal is to predict the target variable $\hat{y}(\vec{x})$ using Eqn. 4.1. The proposed recommendation module is formulated as a sparse matrix. The rows of the matrix are generated by concatenating the embeddings of a user feature vector, POI feature vector, and context vector. We consider the check-in flag as the target variable for each row. The proposed model is operated with the following objective function:

$$\hat{y}(\vec{x}) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < \vec{v}_i, \vec{v}_j > x_i x_j, \tag{4.1}$$

where w_0 is the global bias of all user-POI-context tuples, $\vec{x}$ is a concatenation of user feature vector, POI feature vector, and context vector, n is the size of input variables, $< \vec{v}_i, \vec{v}_j >= \sum_{f=1}^{k} v_{i,f}.v_{j,f}$, and k is the dimensionality of factorization. The Factorization Machine can learn latent factors for all the variables, and can also allow the interactions between all pairs of variables. This makes them an ideal candidate to model complex relationships in the data.

4.4 Explanation of recommendation

The POI-aspect bipartite relation derived from Sec. 4.3 is represented as a bipartite graph and the ordered preference of user on aspect categories is extracted and used for explanation of the generated recommendation.

4.4.1 Bipartite Core Extraction (ReEL-Core)

A k-core of a graph is a maximal connected subgraph whose every vertex is connected to at least k other vertices. The k-core analysis is popular for community detection, dense subgraph extraction, and in dynamic graphs. Our method for bipartite core detection is inspired from [Kle99] where each node is assigned two scores - hub score and authority score, which are defined in terms of the outgoing and incoming edges respectively. The hub score (h_i) of a node is proportional to the sum of authority scores of the nodes it links to. The authority score (a_i) of a node is proportional to the sum of hub scores of the nodes it is linked from. Given the initial authority and hub scores of all the nodes, the scores are iteratively updated until the graph converges. For a given user, we consider all the recommended places as the seed nodes and connect them to the aspect nodes for which they have overall positive sentiments (i.e. (no. of positive opinions) $>$ (no. of negative opinions)). This filters out the negatively reviewed places and gives us a bipartite graph as shown in Fig 4.3 (left graph).

We calculate the eigenvectors of the adjacency matrix of the graph to identify the primary eigenpair (largest eigenvalue). The eigenvalue is used as a measure of the density of links in the graph. The iterative algorithm gives the largest eigenvalue (primary eigenpair). The primary eigenpair corresponds to the primary bipartite core (most prevalent set of POI-aspect pairs) and non-primary eigenpairs correspond to the secondary bipartite cores (less prevalent set of POI-aspect pairs). The most dense subgraph (e.g., the right subgraph in Figure 4.3 with nodes AC_1, P_1, P_2, and P_3) is extracted as the primary bipartite core. After finding the primary core, the edges relevant to this core are removed and the process is repeated on residual graph to get the next prevalent bipartite cores. Removal of edges within the primary core will still leave the nodes connected to other aspect nodes which belong to the secondary bipartite cores. The bipartite cores are used in the order (primary, secondary, etc.) when recommendation is generated. The aspects in the bipartite cores are used to explain recommendation of relevant places.

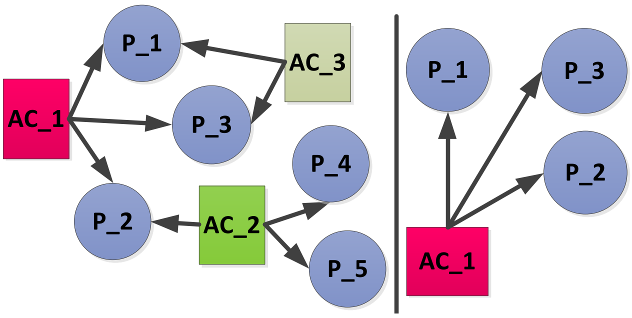


Figure 4.3: Place Aspect Graph (AC_k = aspect k, P_i = places) (Left subgraph is a bipartite graph and the right one is a primary bipartite core)

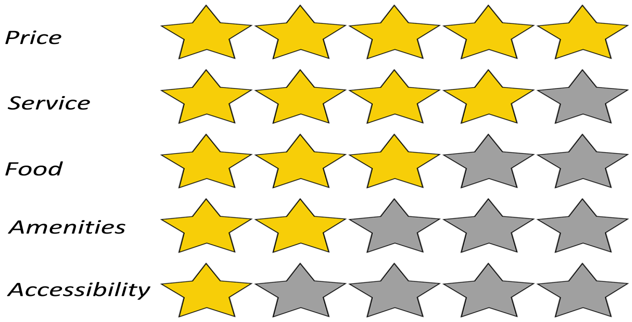


Figure 4.4: Aspect score to star ratings for a POI

Explanation generation: A bipartite core consists of densely connected nodes and resembles the set of place nodes which are mostly known for the relevant aspect nodes. For a user, we generate the POI-aspect relations from the ordered bipartite cores as:

Aspect 1: $POI_1, POI_2, ..., POI_i$

Aspect 2: $POI_i, POI_j, ..., POI_{j+k}$

.....

Aspect k: $POI_1, POI_i, ..., POI_j$,

where each row gives the aspect from the ordered bipartite core and the relevant set of POIs that are popular for that aspect. We also generate the score of each POI_i on each aspect as:

Aspect 1: $\text{Score}_{i,1}$

Aspect 2: $\text{Score}_{i,2}$

.....

Aspect k: $\text{Score}_{i,k}$,

where $\text{Score}_{l,a}$ represents the score of POI_l by the aspect a for all users, and is defined as: $\text{Score}_{l,a}=\sum_{i=1}^{k} \frac{1}{i} * \mid core_{l,a,i} \mid$, where the term $\mid core_{l,a,i} \mid$ represents the number of times the POI_l was in i^{th} bipartite core for the aspect a on all users, and k represents the ordered number of bipartite cores used (e.g., k=1 is for primary bipartite core, k=2 for secondary core, and so on). The scores computed are interpolated to the 5-star rating scheme (see Figure 4.4).

As an example, the review text "Tasty *free* hot *breakfast* and *friendly staffs*", implies that the reviewer cares about the *"Price"* and *"Service"* aspects, and a primary bipartite core for this user should contain these aspects and relevant places. Given the place "Hyatt Regency" and "The Setai Miami Beach" have overall positive opinions for the "Price" aspect, they are included in the primary bipartite core (i.e. related to "Price") and the explanation is generated graphically as shown in Figure 4.4 and is supplemented with text as:

Recommended Place: Hyatt Regency, The Setai Miami Beach, ...;

Explanation: Popular for Price.

4.4.2 Dense subgraph extraction (ReEL-Dense)

This model exploits the weight of user-aspect and place-aspect relation to incorporate the extent of user preferences on aspects and the popularity information of a place through the aspects.

Figure 4.5 shows a basic representation of the network and extraction of dense subgraphs. The POI-aspect edge is weighted by the normalized measure of frequency of overall positive opinions on the aspect for the POI. The user-aspect edge is weighted by the normalized measure of number of times the user reviewed on the aspect. We exploit the random extraction of connected components from the network and proceed with the components having high similarity score. If γ is a random permutation applied on the homogeneous sets A and B (e.g., set A has only user nodes and set B has only aspect nodes), then their similarity score is defined as:

$$\mathrm{Sim}^{\gamma}(A,B) = \frac{f(A,B)}{f(A)+f(B)} \tag{4.2}$$

where $f(A,B) = \sum\limits_{\substack{a\in A, b\in B\\(a,b)\in E}} W_{a,b}$, where $W_{a,b}$ is the weight of edge (a,b) that is normalized to all the edges outgoing from node a, $f(A) = \sum\limits_{(a,i)\in E} W_{a,i}$ is the sum of normalized weights of all edges outgoing from node a, and $f(B) = \sum\limits_{(i,b)\in E} W_{i,b}$ is the sum of normalized weights of all edges incidence on node b. We assume that absence of POI-aspect edge indicates that the place is not known for that aspect (e.g., the aspect is irrelevant). We can use the min-wise independent permutations [BGMZ97, BCFM00] technique to avoid exploitation on each and every permutation to find the sets with high similarity score. We use some predefined number of permutations (c=10) and do not focus on the min-wise independence of the permutations.

Algorithm 2 ShingleFinder(G = (V,E), c, s, k)

```
//G is the input graph, V is the set of vertices, and E is the set of
  edges, c is the number of permutations, s is the length of each set, k is
  the number of shingles to be extracted
initialize L as an empty list
for each place node do
    for j = 1 to c do
        get a set of s aspect nodes
        find aggregated similarity for the place and aspect nodes in this
  set using Eqn. 4.2
        store this set and its score in L
    end for
end for
return k sets with high similarity score (these sets are called shingles)
  from L
```

Algorithm 2 defines shingles extraction process from a bipartite graph. For each POI, we apply Algorithm 2 to find the set of aspect nodes linked to it and extract the k shingles for it. For each shingle, we find the list of all POI nodes that contain it. These are the POIs that are mostly reviewed for the aspect nodes contained in the shingle (see Figure 4.5). As shingles can contain overlapping set of aspects, it can represent the POIs and user preferences of overlapping aspects as well.

The shingles of a user node represent the set of aspects that adhere to her preferences (the preference can be ordered based on the similarity score of a user node to the shingles). As our goal is to cluster (user, POI) tuples, we need to find the sets of user and POI nodes that share sufficiently large number of shingles. Each shingle contains the associated aspects which relates users and POIs. We can easily find the top n_u users and top n_l POIs whose similarity score is high for this shingle.

The overall process can be achieved in polynomial time [BGMZ97, BCFM00] and is dependent on the number of nodes in the graph, number of shingles to use, and the size of a shingle. The normalized similarity score between a POI$_l$

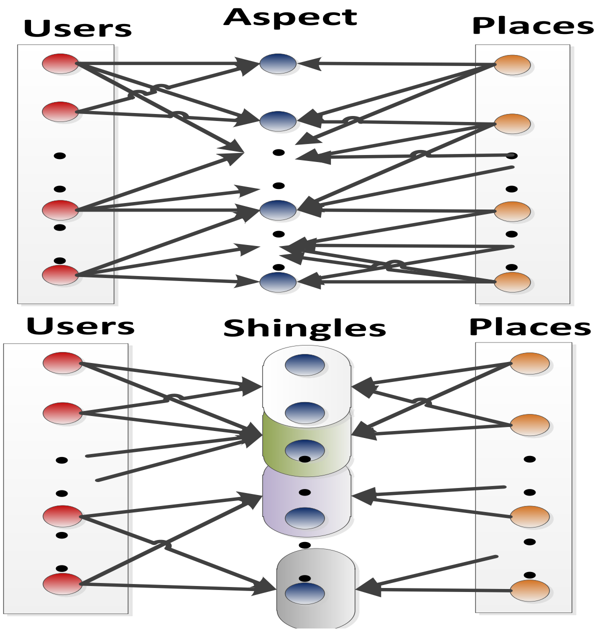


Figure 4.5: Shingles extraction (shown without edge weights)

and an aspect (a) from all shingles is defined as:

$$Score(l,a) = \frac{1}{\mid Sh \mid} \sum_{a \in Sh} \frac{1}{k} sim^{\gamma}(l, Sh), \tag{4.3}$$

where Sh is the set of ordered shingles that contain aspect a, and k is the similarity-based order of the relevant shingle. This score is interpolated to the 5-star rating scheme similar to ReEL-Core.

Finding the subsets of aspects with highest similarity score not only facilitates explanation of recommendation but also provisions clustering of users who have similar preferences on aspects (even in absence of explicit social links) and generating a group recommendation. It can also be used to generate preference wise recommendation (e.g., for the set of users $\{u_1, u_2, u_5\}$ the set of aspects {"food", "service"} might be interesting, for the set of users $\{u_1, u_2, u_3\}$ the set of aspects {"food", "price"} might be interesting, etc.). This can also facilitate the clustering of POIs that are preferred for similar aspects (e.g., the set of hotels that are popular for "service").

4.4.3 Ranking Method (ReEL-Rank)

This model uses the frequency of usage of an aspect to a place. The places recommended to a user and the places' relevant aspects are used as graph nodes. The weight of a place-aspect edge indicates the overall positive opinions on the place for the aspect. A ranking function is then defined as:

$$Rank(i) = \frac{1-d}{N} + d * \sum_{(j,i)\in E} \frac{Rank(j) * W_{j,i}}{O_j}, \tag{4.4}$$

where Rank(i) is the rank of a node i, d (=0.85) is the damping factor, N is number of nodes in the graph, E is set of edges in the graph, $W_{j,i}$ is weight of the edge (j, i), and O_j is number of outgoing links from node j. The ranks are iteratively updated till the graph is converged. The highest ranking aspect node and its highest ranking neighbors give the places that are noted for this aspect. Similarly, other higher ranking aspect nodes and their neighbors are accessed to get the other place-aspect pairs. For a given aspect, the neighbor nodes are sorted based on their rank before the explanation is generated. An explanation of the following form is generated: (i) Food: Places ordered by rank: Place 1, Place 2, ...(ii) Service: Places ordered by rank: Place 4, Place 5, ..., etc. The rank of a place on an aspect is aggregated from all the users to get the star rating score.

4.5 Evaluation

We defined four models: (i) DAP - the model that used a deep network and factorization machine for recommendations and has no provision for explanation,

[1] https://www.yelp.com/dataset_challenge

[2] Wang et al. [WLZ11]

[3] http://insideairbnb.com/get-the-data.html

[4] explicitly missing ratings, neutral, and zero ratings are not shown

Aspect	**Terms**
Price	cash, redeem, cheap, expensive, afford, refund, skyrocket, economize, reimburse, discount
Food	cappuccino, buffet, shell, salami, healthy, mushroom, croissant, cranberry , sushi, broccoli
Pet	mew, swan, cat, fish, ant, pony, dog, bird, duck, purr
Service	friendly, repair, employment, safari, servings, discount, checkouts, cleansing, sightseeing, attitude
Amenities	breakfast, massage, yoga, gabmle, excursion, exercise, sightseeing, housekeeping, exercise, television

Table 4.2: Top-10 terms in different aspect categories

Bipartite Cores	**User u_1**	**User u_2**	**User u_3**
First core	*Price* 103 places	*Service* 137 places	*Price* 272 places
Second core	*Pet* 103 places	*Price* 137 places	*Service* 272 places
Third core	*Service* 47 places	*Pet* 137 places	*Pet* 272 places
Fourth core	*Food* 103 places	*Food* 42 places	*Food* 1 place
Fifth core	*Amenities* 9 places	*Amenities* 137 places	*Amenities* 81 places

Table 4.3: Summary of bipartite cores of three users

Attributes	**Yelp**[1]	**TripAdvisor**[2]	**AirBnB**[3]
Reviews	2,225,213	246,399	570,654
Users	552,339	148,480	472,701
Places	77,079	1,850	26,734
Words	302,979,760	43,273,874	54,878,077
Sentences	18,972,604	2,167,783	284,1004
Avgerage Sentences/review	8.53	8.79	4.98
Avgerage Words/review	136.15	175.62	96.16
Avgerage Reviews/user	4.03	1.66	1.20
Avgerage Reviews/place	28.87	133.18	21.34
4, 5 stars[4]	591,618 and 900,940	78,404 and 104,442	479,842
1, 2 stars	260,492 and 190,048	15,152 and 20,040	5,766

Table 4.4: Statistics of the datasets

(ii) ReEL-Core - the model that used bipartite core, (iii) ReEL-Dense - the model that used dense subgraph extraction, and (iv) ReEL-Rank - the model that used a ranking approach for explanation generation. We also evaluated the Aspect extraction, Aspect categorization, Sentence-aspect classification modules in terms of accuracy.

1. Aspect extraction: We used the SemEval 2014 Task 4: Aspect Based Sentiment Analysis Annotation dataset as the benchmark data and were able to get an accuracy of 70.04%.
2. Aspect categorization: We got an accuracy of 67.12% with the SemEval 2014 Task 4: Aspect Based Sentiment Analysis Annotation dataset.
3. Sentence-aspect classification: We used 100, 150, and 200 epochs with 32 and 64 batches. With 200 epochs and 64 batches, we got 69.01% accuracy on Yelp dataset.

We compared the performance of our proposed models with the following models: (1) UCF [HKBR99] uses the user-based collaborative filtering technique, (2) ICF [SKKR01] uses item-based collaborative filtering, (3) PPR [Hav02] uses personalized page ranking, (4) Guo et al. [GSZ+17] uses aspect-aware POI recommendation, (5) ORec [ZCZ15] uses opinion-based POI recommendation, (6) Word-embedding approach: In this approach, the review sentences from a user and the one for an item are mapped to a latent space using the word embedding [MSC+13]. For a user, the K-nearest neighbors in the space were considered as the top-K recommendations, (7) Latent Dirichlet Allocation approach [BNJ03]: In this model, we extract the topics relevant to a user and the topics relevant to places. The user-place tuples with most common topics are used for the recommendation, and (8) *DeepConn* [ZNY17]: This is the CNN-based model which uses the review embeddings but ignores the other contextual embedding and the polarity of reviews.

4.5.1 Dataset

We used three real-world datasets to evaluate the proposed models. Table 4.4 shows that in all three datasets, most of the users tend to give high (positive) ratings to the places. The top-10 terms of different aspects are illustrated in Table 4.2.

Experimental settings: We used a 5-fold cross validation to evaluate the models. The frequency thresholds for noun and noun phrase extraction were set to 100, 250, and 500. Our experimental analysis show better results with 100. The CNN used 128 filters, 64 batches, 200 epochs, and embedding vectors of size 384. We used an Ubuntu 14.04.5 LTS, 32 GB RAM, a Quadcore Intel(R) Core(TM) i7-3820 CPU @ 3.60 GHz machine. We used the same configuration with Tesla K20c 6 GB GPU to evaluate neural network-based models.

Models	**Precision**	**Recall**	**F-Score**
UCF [HKBR99]	0.23000	0.56800	0.32741
ICF [SKKR01]	0.20100	0.51000	0.28835
PPR [Hav02]	0.23640	0.57000	0.33420
Guo et al. [GSZ$^+$17]	0.52000	0.77420	0.62213
ORec [ZCZ15]	0.50030	0.61000	0.54973
LDA [BNJ03]	0.50160	0.48280	0.49200
Embedding [MMO16]	0.50020	0.71250	0.58780
DeepConn [ZNY17]	0.50510	0.79350	0.61720
DAP	0.61550	0.89630	0.72980
ReEL-Core	**0.71680**	**0.89960**	**0.79780***
ReEL-Rank	0.67740	0.88420	0.76710
ReEL-Dense	0.67310	0.87940	0.76250

Table 4.5: Performance of models (* means statistically significant at 95% confidence interval) in Yelp dataset

Models	**Precision**	**Recall**	**F-Score**
UCF [HKBR99]	0.30000	0.55700	0.38996
ICF [SKKR01]	0.25000	0.52000	0.33766
PPR [Hav02]	0.35000	0.58000	0.43656
Guo et al. [GSZ$^+$17]	0.55000	0.77430	0.64315
ORec [ZCZ15]	0.51000	0.65130	0.57205
LDA [BNJ03]	0.50000	0.79680	0.61440
Embedding [MMO16]	0.57110	0.79710	0.66540
DeepConn [ZNY17]	0.56340	**0.87810**	0.68640
DAP	0.61310	0.79880	0.69370
ReEL-Core	**0.63880**	0.83410	**0.72350***
ReEL-Rank	0.63660	0.81120	0.71330
ReEL-Dense	0.62540	0.79980	0.70190

Table 4.6: Performance of models (* means statistically significant at 95% confidence interval) in TripAdvisor dataset

Models	**Precision**	**Recall**	**F-Score**
UCF [HKBR99]	0.23200	0.56500	0.32893
ICF [SKKR01]	0.20200	0.50000	0.28775
PPR [Hav02]	0.24700	0.56000	0.34280
Guo et al. [GSZ+17]	0.54000	0.76100	0.63173
ORec [ZCZ15]	0.52700	0.60200	0.56201
LDA [BNJ03]	0.50000	0.59480	0.54330
Embedding [MMO16]	0.61640	0.62430	0.62030
DeepConn [ZNY17]	0.60010	0.68320	0.63890
DAP	0.59720	0.78450	0.67810
ReEL-Core	**0.62160**	**0.81830**	**0.70650***
ReEL-Rank	0.61610	0.80730	0.69880
ReEL-Dense	0.60770	0.79700	0.68960

Table 4.7: Performance of models (* means statistically significant at 95% confidence interval) in Airbnb dataset

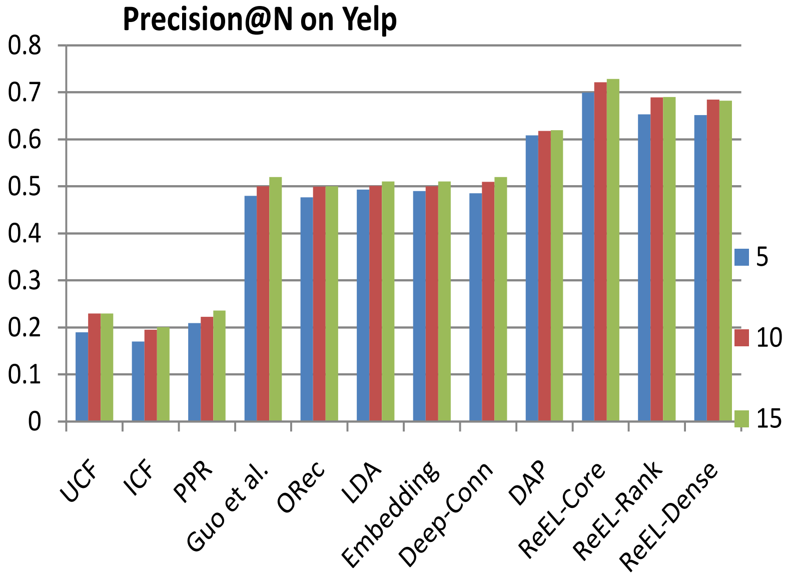


Figure 4.6: Precion@N for Yelp dataset

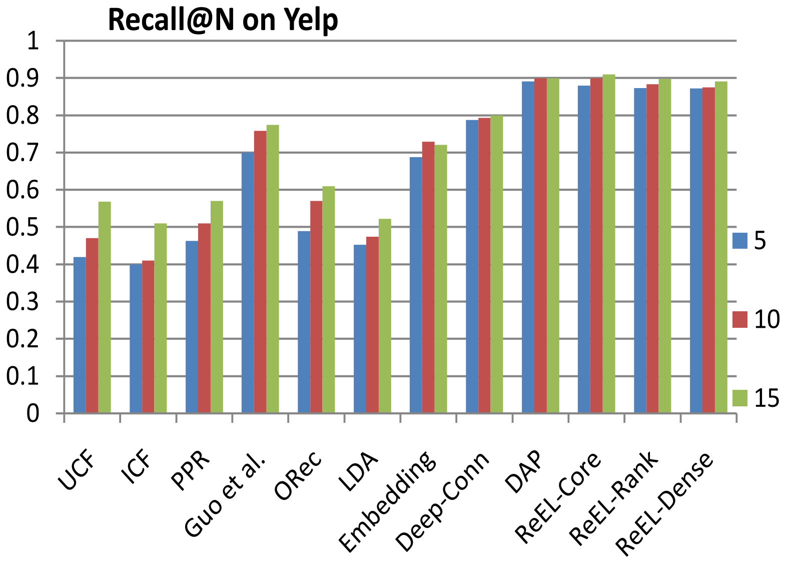


Figure 4.7: Recall@N for Yelp dataset

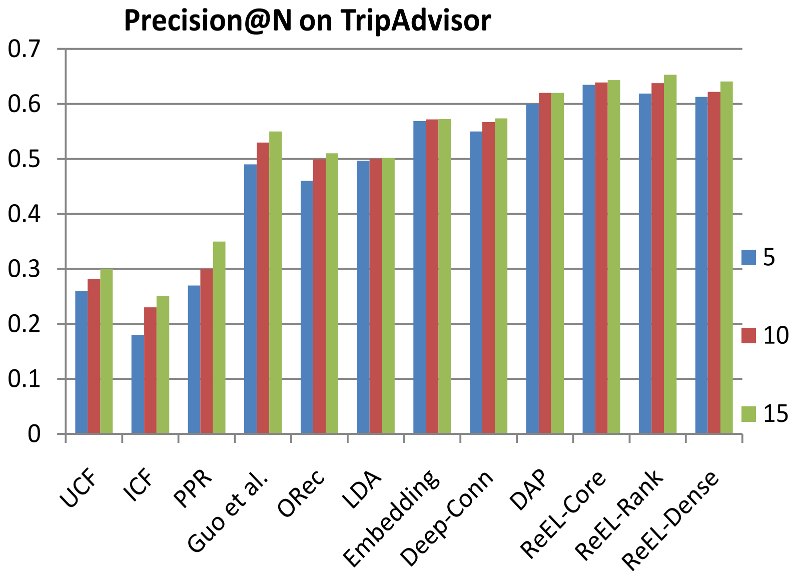


Figure 4.8: Precion@N for TripAdvisor dataset

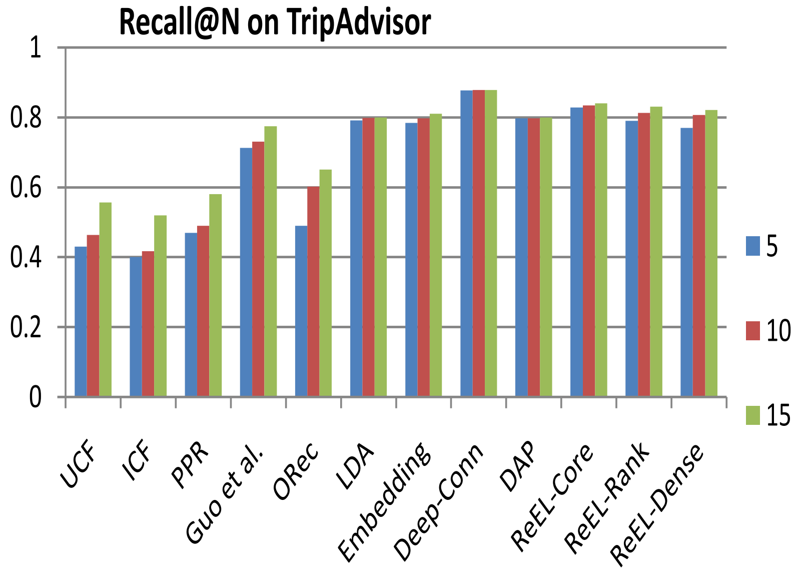


Figure 4.9: Recall@N for TripAdvisor dataset

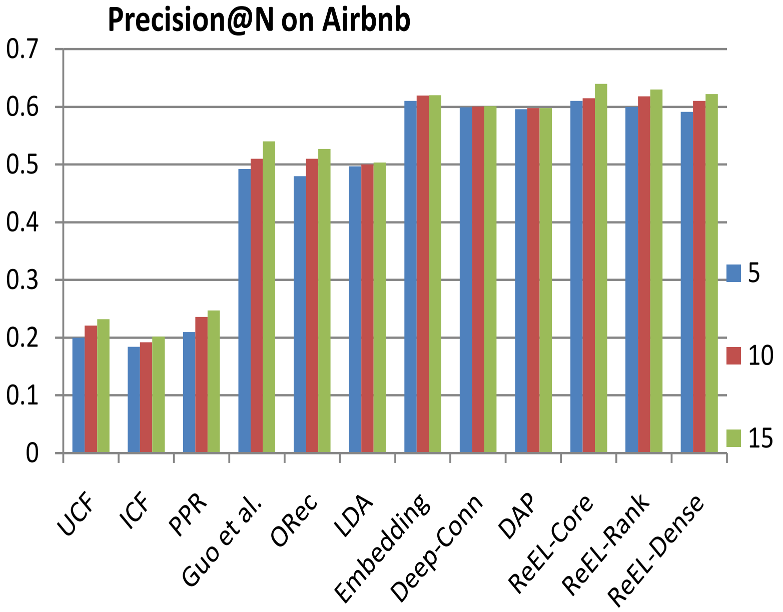


Figure 4.10: Precion@N for Airbnb dataset

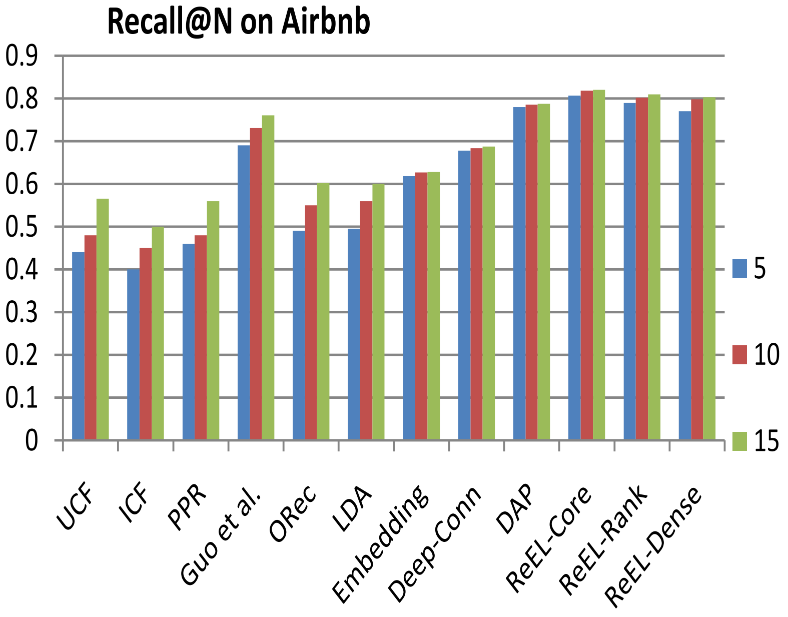


Figure 4.11: Recall@N for Airbnb dataset

4.5.2 Experimental Results and Discussion

We used the reviews of users and places with at least five reviews. We used a 5-fold cross validation and the precision (p), recall (r), and f-score (2*p*r/(p+r)) metrics for evaluation. We considered the top @5, @10, @15, and @20 recommended items for the evaluation. The evaluation of different models is shown in Table 4.5 - 4.7. The Precision@N and Recall@N of different models is shown in Figure 4.6 - 4.11.

The results show that the ICF performed least, UCF and PPR performed on par, model from Guo et al. [GSZ$^+$17] performed better than ORec [ZCZ15], LDA [BNJ03], and Embedding [MMO16] models. Among the ones without explanation, DAP performed best on the Yelp dataset. Though it outperformed in other two datasets as well, the difference was not significant. This implies that for larger datasets, the performance of the proposed model is outstanding. This is common with DNNs which need a reasonably large training data for better performance. The recall of DeepConn [ZNY17] was higher than that of **DAP** in the TripAdvisor dataset but its precision was lower. This might be because of the sentence-level sentiment which was exploited in DAP but not in DeepConn [ZNY17].

Unlike DAP, which provided a single list of recommendations and selected top@N POIs from the list, the **ReEL-Core** and **ReEL-Rank** produced individual lists for each aspect, and outperformed DAP because they categorized recommendations into different aspect categories which led to the re-ordering of the items into small recommendation lists. This re-ordering can help increase the number of true positives and decrease the false positives, as the least preferred items might move to the later part of the recommended lists and the more preferred ones move to the front part of the lists. The ReEL-Core outperformed ReEL-Rank and ReEL-Dense. One reason is due to the repeated bipartite core extraction by ReEL-Core where the nodes got re-ranked for every bipartite core but the ReEL-Rank only ranked all the nodes just once. After having the ordered set of places within each aspect, having an explanation of type similar to [LMRS15] (i.e. place A is better than 80% of places for "Food", etc.) can be achieved by counting the number of places behind the target place in the recommended list.

4.5.3 Evaluation of Explainability

For a place p, the aspect popularity of an aspect a can be defined in terms of the number of positive and negative mentions:

$$AspectPopularity(p_a) = \sum_{sentence \in Review_p} (\mid positive \mid a - \mid negative \mid a). \qquad (4.5)$$

To check the presence of correct aspects in the explanation, we ordered the aspects of every place based on the aspect popularity score. We used a trigram across the extracted aspects to identify the sentiment polarity of the aspects. The relevant aspects were ordered by the aspect popularity score. So, a place can be represented by the set of aspects ordered by the popularity: $p_a =$

$\{a_1, a_2, .., a_n\}$. For every explanation, we took the aspects for which a place was recommended. The aspects were ordered based on the order of cores (primary, secondary, etc.). This gave us another set of aspects for every place. The performance of explainability was then measured in terms of Levenshtein distance between the lists. The average Levenshtein distance across all places was observed to be 20%.

4.5.4 Impact of Explanation - A Case Study

We analyzed the role of ReEL-Core using top-5 bipartite cores (see Table 4.3) extracted for users - "7iigQ2XM-V0ciwmCIdrIBA", "7Mg6r6g7RUwQH_Bllrd-wQ", and "9HDElil2309UajBgtYcD4w", hereafter called as u_1 and u_2, and u_3 respectively. We can see that the ordered preferences of user u_1 are "Price", "Pet", "Service", "Food", and "Amenities". This implies the highest preference of u_1 on "Price", regardless of the order of POIs recommended.

For user u_1, the POI "NK3S3U6TQtysH_-eqT3bBQ" was the second highly recommended place by regular recommender. With the **ReEL-Core**, it is categorized into "Others" bipartite core - *the sixth core*. If the user really cares about other cores (i.e. related to other aspect categories) then having it in sixth core is better than having it in front list. The least recommended POI "p9Bl3BxPltz2WnIxJLnBvw" by simple recommender is now categorized as the least popular item for the primary bipartite core (i.e. related to "Price"), and three other secondary cores (i.e. related to "Service", "Pet", and "Food"). Many POIs ranked in the later part of the list by the simple recommenders are found within top-20 of the different bipartite cores. Have this user used the simple recommendation, and considered only the top-20 recommendations, then these items would have been missed. A sample explanation for user u_1 is the ordered set of places taken from the ordered bipartite cores:

Recommendation: (1) Place 1, Place 2,...; **Explanation**: Popular for Price. (2) Place 3, Place 4,...; **Explanation**: Popular for Service.

Similarly, the place "v4iA8kusUrB19y2QNOiUbw" that was most recommended item for user u_2 by the simple recommender is categorized to sixth bipartite core (i.e. "Others"). The place "HxPpZSY6Q1eARuiahhra6A" that did not fit in top-20 of simple recommender is found in the sixth position of first three bipartite cores. The location "mh1le9QGMrZLohAjfheJJg" which was the second least recommended by simple recommender is categorized as the second least preferred item for the first five bipartite core (i.e. "Service", "Price", "Pet", "Food", and "Amenities"). A similar analysis observed for 500 other users is skipped due to space constraint.

4.6 Conclusion and Future Work

We formulated user-aspect bipartite relation as a bipartite graph and exploited bipartite-core, shingles, and ranking-based techniques to predict the ordered aspect preferences of users for explainable recommendation. The proposed models supplemented with explanations outperformed the ones without explanation, and gained significant improvement (e.g., 5.8% to 29.5% from DeepConn [ZNY17], and 11.1% to 27.4% from Guo et al. [GSZ$^+$17]) on F-score over relevant studies. In future, we would like to exploit different aspect extraction techniques, cluster the users based on their preference order on aspect categories, and generate recommendations for a group.

CHAPTER 5

CONCLUSION AND FUTURE DIRECTIONS

The role of contexts is inevitable for an efficient recommendation. This dissertation developed machine learning models for multi-context Point-of-Interest recommendation, inspired from personalized ranking, non-negative matrix factorization, hierarchical clustering, and neural networks. It defined three main research problems and presented machine learning models to solve those problems. The core contributions of the research are as follows:

1. Analysis of roles of major contexts (categorical, temporal, spatial, and social) in POI recommendation and incorporation of the major contexts into a single recommendation model.
2. Formulation of user activity and location influence into the POI recommendation problem and modeling user preference as hierarchical structure and aggregating the hierarchies to represent the aggregated locality preferences.
3. Exploitation of user reviews to extract aspect-opinion correlation and to generate aspect-aware explanation for POI recommendation.

The context-aware recommendation is an emerging research area. As a follow up to the work presented in this dissertation, some potential directions are identified:

1. We exploited the user preferences as a hierarchy and aggregated the hierarchies to represent locality preferences. The hierarchies can also be potentially adapted to many research directions, such as knowledge discovery, question answering, and so forth.
2. The sequence recommendation problem can be another direction to explore. This is useful in many real-world problems, such as itinerary planning and tour recommendations. The check-in behavior of users can

be chronologically sorted and their sequential check-in trend can be exploited using sequence models, such as Recurrent Neural Network (RNN), Long short-term memory (LSTM), and Gated Recurrent Unit (GRU) to learn the sequence and generate sequence recommendation.

3. The fairness attribute can be another interesting direction to explore. It can be interesting to know if all users are fairly recommended the relevant items and if all items are fairly recommended to the relevant users. This research direction can be further exploited on reciprocal recommendation where both ends are homogeneous entities (e.g., user-user network in dating partner recommendation system).

BIBLIOGRAPHY

[AK17] Avi Arampatzis and Georgios Kalamatianos. Suggesting points-of-interest via content-based, collaborative, and hybrid fusion methods in mobile devices. *ACM Transactions on Information Systems (TOIS)*, 36(3):23, 2017.

[AT11] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011.

[BCFM00] Andrei Z Broder, Moses Charikar, Alan M Frieze, and Michael Mitzenmacher. Min-wise independent permutations. *Journal of Computer and System Sciences*, 60(3):630–659, 2000.

[BG08] Christos Boutsidis and Efstratios Gallopoulos. Svd based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition*, 41(4):1350–1362, 2008.

[BGMZ97] Andrei Z Broder, Steven C Glassman, Mark S Manasse, and Geoffrey Zweig. Syntactic clustering of the web. *Computer Networks and ISDN Systems*, 29(8-13):1157–1166, 1997.

[BILB18] Ramesh Baral, S.S Iyengar, Tao Li, and N Balakrishnan. Close: Contextualized location sequence recommender. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 470–474. ACM, 2018.

[BILZ18] Ramesh Baral, S.S Iyengar, Tao Li, and XiaoLong Zhu. Hicaps: Hierarchical contextual poi sequence recommender. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 436–439. ACM, 2018.

[BIM19] Ramesh Baral, SS Iyengar, and Asad M Madni. Interdependent networks from societal perspective: Mits (multi-context. *Sustainable Interdependent Networks II: From Smart Power Grids to Intelligent Transportation Networks*, page 1, 2019.

[BL97] Michael J Berry and Gordon Linoff. *Data mining techniques: for marketing, sales, and customer support.* John Wiley & Sons, Inc., 1997.

[BL16] Ramesh Baral and Tao Li. Maps: A multi aspect personalized poi recommender system. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 281–284. ACM, 2016.

[BL17] Ramesh Baral and Tao Li. Exploiting the roles of aspects in personalized poi recommender systems. *Data Mining and Knowledge Discovery*, pages 1–24, 2017.

[BM05] Mustafa Bilgic and Raymond J Mooney. Explaining recommendations: Satisfaction vs. promotion. In *Beyond Personalization Workshop, IUI*, volume 5, page 153, 2005.

[BMPW98] Sergey Brin, Rajeev Motwani, Lawrence Page, and Terry Winograd. What can you do with a web in your pocket? *IEEE Data Eng. Bull.*, 21(2):37–47, 1998.

[BNJ03] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[BS01] Keith Bradley and Barry Smyth. Improving recommendation diversity. In *Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science, Maynooth, Ireland*, pages 85–94, 2001.

[BWLC16] Ramesh Baral, Dingding Wang, Tao Li, and Shu-Ching Chen. Geotecs: exploiting geographical, temporal, categorical and social aspects for personalized poi recommendation. In *Information Reuse and Integration (IRI), 2016 IEEE 17th International Conference on*, pages 94–101. IEEE, 2016.

[BZIL18] Ramesh Baral, XiaoLong Zhu, S.S Iyengar, and Tao Li. Reel: Review aware explanation of location recommendation. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, pages 23–32. ACM, 2018.

[BZM12] Jie Bao, Yu Zheng, and Mohamed F Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th international conference on advances in geographic information systems*, pages 199–208. ACM, 2012.

[CAS16] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 191–198. ACM, 2016.

[CBdG$^+$17] Annalina Caputo, Pierpaolo Basile, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, and Gaetano Rossiello. Sabre: A sentiment aspect-based retrieval engine. In *Information Filtering and Retrieval*, pages 63–78. Springer, 2017.

[CC15] Guanliang Chen and Li Chen. Augmenting service recommender systems by incorporating contextual opinions from user reviews. *User Modeling and User-Adapted Interaction*, 25(3):295–329, 2015.

[CML11] Eunjoon Cho, Seth A Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090. ACM, 2011.

[COX16] Dawei Chen, Cheng Soon Ong, and Lexing Xie. Learning points and routes to recommend trajectories. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 2227–2232. ACM, 2016.

[CT12] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[CW17] Li Chen and Feng Wang. Explaining recommendations based on feature sentiments in product reviews. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 17–28. ACM, 2017.

[CWB$^+$11] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[CYKL12] Chen Cheng, Haiqin Yang, Irwin King, and Michael R Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[CYLK13] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, volume 13, pages 2605–2611, 2013.

[CZZ$^+$17] Jiajun Cheng, Shenglin Zhao, Jiani Zhang, Irwin King, Xin Zhang, and Hui Wang. Aspect-level sentiment classification with heat (hierarchical attention) network. In *Proceedings of the 2017 ACM* *on Conference on Information and Knowledge Management*, pages 97–106. ACM, 2017.

[DHX$^+$06] Chris Ding, Xiaofeng He, Hui Xiong, Hanchuan Peng, and Stephen R Holbrook. Transitive closure and metric inequality of weighted graphs: detecting protein interaction modules using cliques. *International journal of data mining and bioinformatics*, 1(2):162–177, 2006.

[DJLS07] Chris Ding, Rong Jin, Tao Li, and Horst D. Simon. A learning framework using green's function and kernel regularization with application to recommender system. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 260–269. ACM, 2007.

[DS17] Ruihai Dong and Barry Smyth. User-based opinion-based recommendation. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4821–4825. AAAI Press, 2017.

[Fel98] Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998.

[FYL13] Gregory Ference, Mao Ye, and Wang-Chien Lee. Location recommendation for out-of-town users in location-based social networks. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 721–726. ACM, 2013.

[GAN15] Jean-Benoît Griesner, Talel Abdessalem, and Hubert Naacke. Poi recommendation: Towards fused matrix factorization with geographical and temporal influences. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 301–304. ACM, 2015.

[GGJ11] Fatih Gedikli, Mouzhi Ge, and Dietmar Jannach. Understanding recommendations by reading the clouds. In *International Conference on Electronic Commerce and Web Technologies*, pages 196–208. Springer, 2011.

[GJG14] Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. How should i explain? a comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies*, 72(4):367–382, 2014.

[GKR98] David Gibson, Jon Kleinberg, and Prabhakar Raghavan. Inferring web communities from link topology. In *Proceedings of the* *ninth ACM conference on Hypertext and hypermedia: links, objects, time and space—structure in hypermedia systems: links, objects, time and space—structure in hypermedia systems*, pages 225–234. ACM, 1998.

[GLX$^+$11] Yong Ge, Qi Liu, Hui Xiong, Alexander Tuzhilin, and Jian Chen. Cost-aware travel tour recommendation. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 983–991. ACM, 2011.

[GNOT92] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[GSZ$^+$17] Qing Guo, Zhu Sun, Jie Zhang, Qi Chen, and Yin-Leng Theng. Aspect-aware point-of-interest recommendation with geo-social influence. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 17–22. ACM, 2017.

[GTHL13] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 93–100. ACM, 2013.

[GTL12] Huiji Gao, Jiliang Tang, and Huan Liu. Exploring social-historical ties on location-based social networks. In *ICWSM*, 2012.

[GZC$^+$09] Ido Guy, Naama Zwerdling, David Carmel, Inbal Ronen, Erel Uziel, Sivan Yogev, and Shila Ofek-Koifman. Personalized recommendation of social software items based on social relations. In *Proceedings of the third ACM conference on Recommender systems*, pages 53–60. ACM, 2009.

[Hav02] Taher H Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM, 2002.

[HCKC15] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1661–1670. ACM, 2015.

[HE13] Bo Hu and Martin Ester. Spatial topic modeling in online social media for location recommendation. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 25–32. ACM, 2013.

[HJ97] Pierre Hansen and Brigitte Jaumard. Cluster analysis and mathematical programming. *Mathematical programming*, 79(1-3):191–215, 1997.

[HJE13] Bo Hu, Mohsen Jamali, and Martin Ester. Spatio-temporal topic modeling in mobile social media for location recommendation. In *2013 IEEE 13th International Conference on Data Mining*, pages 1073–1078. IEEE, 2013.

[HKBR99] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM, 1999.

[HKR00] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250. ACM, 2000.

[HKV08] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. IEEE, 2008.

[HPK11] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[HSL14] Longke Hu, Aixin Sun, and Yong Liu. 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*, pages 345–354. ACM, 2014.

[JO11] Yohan Jo and Alice H Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 815–824. ACM, 2011.

[JQMF16] Shuhui Jiang, Xueming Qian, Tao Mei, and Yun Fu. Personalized travel sequence recommendation on multi-source big social media. *IEEE Transactions on Big Data*, 2(1):43–56, 2016.

[JS02] Martin Szummer Tommi Jaakkola and Martin Szummer. Partially labeled classification with markov random walks. *Advances in neural information processing systems (NIPS)*, 14:945–952, 2002.

[JSW$^+$12] Zhaoyan Jin, Dianxi Shi, Quanyuan Wu, Huining Yan, and Hua Fan. Lbsnrank: personalized pagerank on location-based social networks. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 980–987. ACM, 2012.

[KBV09] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[Kim14] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[Kle99] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[Kor10] Yehuda Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

[LCLK17] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. *Knowledge and Information Systems*, pages 1–32, 2017.

[LCX$^+$14] Qi Liu, Enhong Chen, Hui Xiong, Yong Ge, Zhongmou Li, and Xiang Wu. A cocktail approach for travel package recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 26(2):278–293, 2014.

[LD13] Tao Li and Chris Ding. Non-negative matrix factorizations for clustering: A survey. *Data Clustering: Algorithms and Applications, Data Mining and Knowledge Discovery Series*, page 148, 2013.

[LFYX13] Bin Liu, Yanjie Fu, Zijun Yao, and Hui Xiong. Learning geographical preferences for point-of-interest recommendation. In *Proceedings of the 19th ACM SIGKDD international conference* *on Knowledge discovery and data mining*, pages 1043–1051. ACM, 2013.

[LLAM13] Xin Liu, Yong Liu, Karl Aberer, and Chunyan Miao. Personalized point-of-interest recommendation by mining users' preference transition. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 733–738. ACM, 2013.

[LLL$^+$16] Yanchi Liu, Chuanren Liu, Bin Liu, Meng Qu, and Hui Xiong. Unified point-of-interest recommendation with temporal interval assessment. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1015–1024. ACM, 2016.

[LMRS15] Aonghus Lawlor, Khalil Muhammad, Rachael Rafter, and Barry Smyth. Opinionated explanations for recommendation systems. In *Research and Development in Intelligent Systems XXXII*, pages 331–344. Springer, 2015.

[LSEM12] Justin J Levandoski, Mohamed Sarwat, Ahmed Eldawy, and Mohamed F Mokbel. Lars: A location-aware recommender system. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 450–461. IEEE, 2012.

[LWWT16] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*, pages 194–200, 2016.

[LWZ12] Shoushan Li, Rongyang Wang, and Guodong Zhou. Opinion target extraction using a shallow semantic parsing framework. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1671–1677. AAAI Press, 2012.

[LX13] Bin Liu and Hui Xiong. Point-of-interest recommendation in location based social networks with topic and location awareness. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 396–404. SIAM, 2013.

[LXP$^+$15] Bin Liu, Hui Xiong, Spiros Papadimitriou, Yanjie Fu, and Zijun Yao. A general geographical probabilistic factor model for point of interest recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 27(5):1167–1179, 2015.

[LZX$^+$14] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 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*, pages 831–840. ACM, 2014.

[ME10] Samaneh Moghaddam and Martin Ester. Opinion digger: an unsupervised opinion miner from unstructured product reviews. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1825–1828. ACM, 2010.

[ME11] Samaneh Moghaddam and Martin Ester. Ilda: interdependent lda model for learning latent aspects and their ratings from online product reviews. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 665–674. ACM, 2011.

[ML13] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM, 2013.

[MLS16] Khalil Ibrahim Muhammad, Aonghus Lawlor, and Barry Smyth. A live-user study of opinionated explanations for recommender systems. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, pages 256–260. ACM, 2016.

[MMO16] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. Modelling user preferences using word embeddings for context-aware venue recommendation. *arXiv preprint arXiv:1606.07828*, 2016.

[MNL$^+$16] Cataldo Musto, Fedelucio Narducci, Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Explod: A framework for explaining recommendations based on the linked open data cloud. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 151–154. ACM, 2016.

[MR10] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms.* 2010.

[MSB$^+$14] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[PD84] Janos Podani and TA Dickinson. Comparison of dendrograms: a multivariate approach. *Canadian Journal of Botany*, 62(12):2765–2778, 1984.

[Ren12] Steffen Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57, 2012.

[RFGST09] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 452–461. AUAI Press, 2009.

[RHW86] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.

[RIS+94] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

[RW13] Seyyed Mohammadreza Rahimi and Xin Wang. Location recommendation based on periodicity of human activities and location categories. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 377–389. Springer, 2013.

[SB88] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[SKKR01] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In

Proceedings of the 10th international conference on World Wide Web, pages 285–295. ACM, 2001.

[SM95] Upendra Shardanand and Pattie Maes. Social information filtering: algorithms for automating word of mouth. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217. ACM Press/Addison-Wesley Publishing Co., 1995.

[SMDM16] Torin Stepan, Jason M Morawski, Scott Dick, and James Miller. Incorporating spatial, temporal, and social context in recommendations for location-based social networks. *IEEE Transactions on Computational Social Systems*, 3(4):164–175, 2016.

[SNM08] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Providing justifications in recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 38(6):1262–1272, 2008.

[SWLO09] Bo Shao, Dingding Wang, Tao Li, and Mitsunori Ogihara. Music recommendation based on acoustic features and user access patterns. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(8):1602–1611, 2009.

[TM07] Nava Tintarev and Judith Masthoff. A survey of explanations in recommender systems. In *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, pages 801–810. IEEE, 2007.

[TM08] Ivan Titov and Ryan T McDonald. A joint model of text and aspect ratings for sentiment summarization. In *ACL*, volume 8, pages 308–316. Citeseer, 2008.

[TM12] Nava Tintarev and Judith Masthoff. Evaluating the effectiveness of explanations for recommender systems. *User Modeling and User-Adapted Interaction*, 22(4):399–439, 2012.

[Tob70] Waldo R Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*, pages 234–240, 1970.

[TSK06] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. Introduction to data mining. 2006.

[VSR09] Jesse Vig, Shilad Sen, and John Riedl. Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 47–56. ACM, 2009.

[WHZ$^+$16] Yequan Wang, Minlie Huang, Li Zhao, et al. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615, 2016.

[WLC$^+$16] Xiaoting Wang, Christopher Leckie, Jeffrey Chan, Kwan Hui Lim, and Tharshan Vaithianathan. Improving personalized trip recommendation by avoiding crowds. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 25–34. ACM, 2016.

[WLWK08] Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems (TOIS)*, 26(3):13, 2008.

[WLZ11] Hongning Wang, Yue Lu, and ChengXiang Zhai. Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 618–626. ACM, 2011.

[WPDX17] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *AAAI*, pages 3316–3322, 2017.

[WTM13] Hao Wang, Manolis Terrovitis, and Nikos Mamoulis. Location recommendation in location-based social networks using user check-in data. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 374–383. ACM, 2013.

[WTWL15] Suhang Wang, Jiliang Tang, Yilin Wang, and Huan Liu. Exploring implicit hierarchical structures for recommender systems. In *IJCAI*, pages 1813–1819, 2015.

[WWT$^+$17] Suhang Wang, Yilin Wang, Jiliang Tang, Kai Shu, Suhas Ranganath, and Huan Liu. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *Proceedings of the 26th International Conference on World Wide Web*, pages 391–400. International World Wide Web Conferences Steering Committee, 2017.

[WYC$^+$15] Weiqing Wang, Hongzhi Yin, Ling Chen, Yizhou Sun, Shazia Sadiq, and Xiaofang Zhou. Geo-sage: A geographical sparse additive generative model for spatial item recommendation. In *Pro-*

ceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1255–1264. ACM, 2015.

[WZN+15] Xiangyu Wang, Yi-Liang Zhao, Liqiang Nie, Yue Gao, Weizhi Nie, Zheng-Jun Zha, and Tat-Seng Chua. Semantic-based location recommendation with multimodal venue semantics. *IEEE Transactions on Multimedia*, 17(3):409–419, 2015.

[XLLZ17] Bin Xia, Tao Li, Qianmu Li, and Hong Zhang. Noise-tolerance matrix completion for location recommendation. *Data Mining and Knowledge Discovery*, pages 1–24, 2017.

[XLLZ18] Bin Xia, Tao Li, Qianmu Li, and Hong Zhang. Noise-tolerance matrix completion for location recommendation. *Data Mining and Knowledge Discovery*, 32(1):1–24, 2018.

[XNL+17] Bin Xia, Zhen Ni, Tao Li, Qianmu Li, and Qifeng Zhou. Vrer: context-based venue recommendation using embedded space ranking svm in location-based social network. *Expert Systems with Applications*, 83:18–29, 2017.

[XYW+16] Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, and Sen Wang. Learning graph-based poi embedding for location-based recommendation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 15–24. ACM, 2016.

[YCM+13] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Time-aware point-of-interest recommendation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 363–372. ACM, 2013.

[YSC+13] Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. Lcars: a location-content-aware recommender system. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 221–229. ACM, 2013.

[YXYG16] Zhiwen Yu, Huang Xu, Zhe Yang, and Bin Guo. Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints. *IEEE Transactions on Human-Machine Systems*, 46(1):151–158, 2016.

[YYL10] Mao Ye, Peifeng Yin, and Wang-Chien Lee. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 458–461. ACM, 2010.

[YYLL11] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. 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*, pages 325–334. ACM, 2011.

[YZS$^+$15] Hongzhi Yin, Xiaofang Zhou, Yingxia Shao, Hao Wang, and Shazia Sadiq. Joint modeling of user check-in behaviors for point-of-interest recommendation. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1631–1640. ACM, 2015.

[YZYW13] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhu Wang. A sentiment-enhanced personalized location recommendation system. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, pages 119–128. ACM, 2013.

[ZC15] Jia-Dong Zhang and Chi-Yin Chow. Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 443–452. ACM, 2015.

[ZCXM09] Yu Zheng, Yukun Chen, Xing Xie, and Wei-Ying Ma. Geolife2. 0: a location-based social networking service. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pages 357–358. IEEE, 2009.

[ZCYZ15] Kaiqi Zhao, Gao Cong, Quan Yuan, and Kenny Q Zhu. Sar: A sentiment-aspect-region model for user preference analysis in geotagged reviews. In *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*, pages 675–686. IEEE, 2015.

[ZCZ15] Jia-Dong Zhang, Chi-Yin Chow, and Yu Zheng. Orec: An opinion-based point-of-interest recommendation framework. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1641–1650. ACM, 2015.

[ZL14] Lei Zhang and Bing Liu. Aspect and entity extraction for opinion mining. In *Data mining and knowledge discovery for big data*, pages 1–40. Springer, 2014.

[ZLD10] Li Zheng, Tao Li, and Chris Ding. Hierarchical ensemble clustering. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 1199–1204. IEEE, 2010.

[ZLHL13] Li Zheng, Lei Li, Wenxing Hong, and Tao Li. Penetrate: Personalized news recommendation using ensemble hierarchical clustering. *Expert Systems with Applications*, 40(6):2127–2136, 2013.

[ZLWS15] Chenyi Zhang, Hongwei Liang, Ke Wang, and Jianling Sun. Personalized trip recommendation with poi availability and uncertain traveling time. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 911–920. ACM, 2015.

[ZLXJ11] Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. Clustering product features for opinion mining. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 347–354. ACM, 2011.

[ZLZ$^+$14] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92. ACM, 2014.

[ZNY17] Lei Zheng, Vahid Noroozi, and Philip S Yu. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 425–434. ACM, 2017.

[ZTZX14] Mi Zhang, Jie Tang, Xuchen Zhang, and Xiangyang Xue. Addressing cold start in recommender systems: A semi-supervised co-training algorithm. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 73–82. ACM, 2014.

[ZXL$^+$17] Pengpeng Zhao, Xiefeng Xu, Yanchi Liu, Ziting Zhou, Kai Zheng, Victor S Sheng, and Hui Xiong. Exploiting hierarchical structures for poi recommendation. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 655–664. IEEE, 2017.

[ZYHW16] Wei Zhang, Quan Yuan, Jiawei Han, and Jianyong Wang. Collaborative multi-level embedding learning from reviews for rating prediction. In *IJCAI*, pages 2986–2992, 2016.

VITA

RAMESH BARAL

11200 SW 8th St ECS 354, Miami FL 33199
Email: rbara012@fiu.edu

2013 - Present	M.S. and Ph.D., Computer Science Florida International University, Miami, Florida
2003 - 2007	B.E., Software Engineering Pokhara University, Pokhara, Nepal

PUBLICATIONS AND PRESENTATIONS

- Baral, Ramesh, S. S. Iyengar, Tao Li, and XiaoLong Zhu. “HiCaPS: Hierarchical Contextual POI Sequence Recommender.” In *Proceedings of 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 470-474. ACM, 2018.

- Baral, Ramesh, Sundararaja S. Iyengar, Asad M. Madni (2018). Sustainable Interdependent Networks from Societal Perspective, MITS (Multi-context Influence Tracking on Social Networks) In *Amini, M. Hadi, Kianoosh G. Boroojeni, Sundararaja S. Iyengar, Panos M. Pardalos, Frede Blaabjerg, and Asad M. Madni, (Eds.). Sustainable Interdependent Networks II From Smart Power Grids to Intelligent Transportation Networks.* Vol. 145. Springer, 2018.

- Baral, Ramesh, S. S. Iyengar, Tao Li, and N. Balakrishnan. “CLoSe: Contextualized Lo cation Sequence Recommender.” In *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 470-474. ACM, 2018.

- Baral, Ramesh, XiaoLong Zhu, S. S. Iyengar, and Tao Li. “ReEL: Review Aware Explanation of Location Recommendation.” In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, ACM, 2018.

- Wubai Zhou, Wei Xue, Baral Ramesh, Qing Wang, Chunqiu Zeng, Tao Li, Jian Xu, Zhen Liu, Larisa Shwartz, Genady Ya. Grabarnik. *STAR: A System for Ticket Analysis and Resolution.* In *Proceedings of the 23th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2017.

- Baral, Ramesh, and Tao Li. “Exploiting the roles of aspects in personalized POI recommender systems.” In *Data Mining and Knowledge Discovery* 32, no. 2 (2018): 320-343.

- Baral, Ramesh, and Tao Li. "MAPS: A multi aspect personalized poi recommender system." In *Proceedings of the 10th ACM conference on recommender systems (pp. 281-284)*, 2016. ACM.

- Baral, Ramesh, Dingding Wang, Tao Li, and Shu-Ching Chen. "Geotecs: exploiting geographical, temporal, categorical and social aspects for personalized poi recommendation." In *Proceedings of 17th IEEE International Conference on Information Reuse and Integration (IRI)* pp. 94-101. 2016, IEEE.

- Abeyruwan, Saminda, Baral Ramesh, Ugan Yasavur, Christine Lisetti, and Ubbo Visser. "Humanoid robots and spoken dialog systems for brief health interventions." In *2014 AAAI Fall symposium series.* 2014.