

Exploiting Geographical Neighborhood Characteristics for Location Recommendation

Yong Liu[‡] Wei Wei^{†*} Aixin Sun[‡] Chunyan Miao[‡]
[‡]School of Computer Engineering, Nanyang Technological University, Singapore,
{liuy0054@e., axsun@, ascymiao@}ntu.edu.sg
[†]School of Information Systems, Singapore Management University, Singapore,
weiwei8329@gmail.com

ABSTRACT

Geographical characteristics derived from the historical check-in data have been reported effective in improving location recommendation accuracy. However, previous studies mainly exploit geographical characteristics from a user’s perspective, via modeling the geographical distribution of each individual user’s check-ins. In this paper, we are interested in exploiting geographical characteristics from a location perspective, by modeling the geographical neighborhood of a location. The neighborhood is modeled at two levels: the instance-level neighborhood defined by a few nearest neighbors of the location, and the region-level neighborhood for the geographical region where the location exists. We propose a novel recommendation approach, namely Instance-Region Neighborhood Matrix Factorization (IRenMF), which exploits two levels of geographical neighborhood characteristics: a) instance-level characteristics, i.e., nearest neighboring locations tend to share more similar user preferences; and b) region-level characteristics, i.e., locations in the same geographical region may share similar user preferences. In IRenMF, the two levels of geographical characteristics are naturally incorporated into the learning of latent features of users and locations, so that IRenMF predicts users’ preferences on locations more accurately. Extensive experiments on the real data collected from Gowalla, a popular LBSN, demonstrate the effectiveness and advantages of our approach.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Information Filtering

Keywords

Geographical Neighborhood; Location Recommendation; Matrix Factorization; Location-based Social Networks

1. INTRODUCTION

In recent years, we have witnessed the rapid growth and increasing popularity of location-based social networks (LBSNs), e.g.,

*This work was performed while this author was a research fellow at Nanyang Technological University, Singapore.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM’14, November 3–7, 2014, Shanghai, China.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2598-1/14/11 ...\$15.00.
<http://dx.doi.org/10.1145/2661829.2662002>

BrightKite, Foursquare and Gowalla, where users explore their surrounding places and share life experiences via “check-ins”. The available check-in data in LBSNs contain rich knowledge about users’ interests, and thus are beneficial to a wide range of applications such as location recommendation [33], friend recommendation [23], and activity recommendation [34].

As an important application in LBSNs, the personalized location recommender systems (PLRs) can help users explore new locations to enrich their experiences. On the other hand, PLRs can also facilitate third-party developers (e.g., advertisers) to provide more relevant services at the right locations. Therefore, PLRs have drawn great research attention from both academia and industry in recent years [35, 36]. For example, Foursquare, a well-known LBSN that has over 50 million users and 6 billion check-ins (last updated May, 2014), has launched its location recommendation engine since March 2011.¹

The primary idea of PLRs is to predict a user’s preferences on unvisited locations. Among existing PLRs developed for LBSNs, the approaches established on collaborative filtering technique are most widely used to model users’ preferences, according to the historical interactions between users and locations [34]. To improve location recommendation accuracy, auxiliary information such as social information and geographical information, has also been exploited. For instance, social opinions from users’ online social friends [27] or local experts in a new city [2] have been considered for location recommendation. In contrast to social information, geographical information has been found to play a much more important role in the location recommendation task [3].

Prior work that employs geographical information for location recommendation mainly explores geographical characteristics of check-in data from user perspective. Based on the empirical observation that users tend to check-in at nearby locations, the geographical distances between users and locations are considered for location recommendation tasks, using an inverse proportional rule in [17]. To better describe users’ preferences on locations, different methods have been proposed to model users’ check-in behaviors in LBSNs. For example, in [26], a power-law probabilistic model is used to describe the power-law distribution of a user’s check-ins. The approach in [3] utilizes a multi-center Gaussian model to study users’ multi-center check-in behavior. To summarize, existing PLRs only exploit the geographical characteristics from user perspective, via modeling the geographical distribution (e.g., the power-law distribution or multi-center distribution) specific to a particular user’s check-ins.

To the best of our knowledge, geographical characteristics from location perspective have not been exploited for location recom-

¹<http://engineering.foursquare.com/2011/03/22/building-a-recommendation-engine-foursquare-style/>

mentation (see Section 2.3 for details). This kind of characteristics are independent from individual users and should have potential benefits to location recommendation. Motivated by this, in this paper, we propose to study geographical characteristics from location perspective, with an emphasis on the geographical neighborhood.

Through our empirical analysis, we found that nearest neighboring locations tend to share more common visitors. This observation indicates that, in general, a user shares similar preferences on nearest neighboring locations. In our method, this pattern is used as the **instance-level geographical neighborhood characteristics**. Specifically, by introducing a similarity measure, the relationship underlying a user’s preferences on a few nearest neighboring locations is captured and utilized to characterize her preference on a target location.

In addition, our study on check-in data also shows that locations in a geographical region may share similar user preferences. Each geographical region is usually associated with a specific function (e.g., business, entertainment, and education) [4, 29]. Such geographical regions contain additional prior knowledge about the interactions between users and locations. Therefore, we use the geographical region structure as the **region-level geographical neighborhood characteristics**. Specifically, a group lasso penalty is employed to integrate the geographical region structure within the check-in data into the learning of latent factors of users and locations. To summarize, the major contributions made in this paper are as follows:

- We empirically analyze geographical characteristics from location perspective using historical check-in data collected from Gowalla. Two levels (i.e., instance-level and region-level) of geographical neighborhood characteristics have been studied.
- We propose a novel location recommendation framework, i.e., Instance-Region Neighborhood Matrix Factorization (IRenMF), to incorporate aforementioned geographical neighborhood characteristics for improving location recommendation accuracy. In particular, to solve the optimization problem in IRenMF, we propose an alternating optimization strategy, in which an accelerated proximal gradient (APG) method is used to learn the latent factors of locations, considering the geographical region structure of the check-in data.
- We extensively evaluate IRenMF on the datasets collected from Gowalla. Experimental results show that: (1) Compared to the baseline matrix factorization model (i.e., WRMF) without utilizing geographical characteristics, our methods that use either instance-level or region-level geographical neighborhood characteristics significantly improves recommendation accuracy; the improvement is on average by 30.78% and 17.2% respectively, in terms of precision metric P@5. (2) Utilizing both two levels of geographical neighborhood characteristics, IRenMF substantially outperforms two state-of-the-art location recommendation methods, which consider geographical characteristics derived from user perspective.

The rest of this paper is organized as follows. Section 2 reviews the most relevant work about this study. Section 3 introduces our empirical analysis of check-in data and provides some background about matrix factorization. Next, Section 4 presents the details of IRenMF and describes the optimization solution to IRenMF. Then, in Section 5, we report the experimental results conducted on real-world datasets, which demonstrate that the geographical characteristics derived from location perspective can significantly improve location recommendation accuracy. Finally, Section 6 draws the conclusion of this study.

2. RELATED WORK

This study relates with three research areas: collaborative filtering, structured sparsity learning, and personalized location recommendation. Next, we will present an overview of the most related work in each area.

2.1 Collaborative Filtering

The collaborative filtering (CF) technique has been widely used for personalized recommendation tasks [1]. In the literature, there are two main categories of CF methods, namely memory-based CF and model-based CF.

The most popular memory-based CF approaches are user-based CF and item-based CF. The user-based CF predicts a user’s preference on a given item based on the preferences of other similar users on the target item. In contrast, the item-based CF first finds similar items to the target item and then predicts a user’s preference on the target item, according to her preferences on other similar items. In previous work [26,27], the user-based CF has been successfully applied for the location recommendation tasks, and it has been found to significantly outperform item-based CF. Similar results have also been found in our experiments (see Section 5.2.1).

The model-based CF approaches aim to learn an algorithmic model to explain the observed user-item interactions. Most existing model-based CF models deal with rating prediction problems, in which users’ preferences are explicitly indicated by the rating values given to items. However, in most practical scenarios, users’ rating data are unavailable. The personalized recommendations are based on users’ implicit feedback, e.g., purchase history [20], click history [5], and check-in history [3]. In [8], Hu et al. introduced the first attempts applying model-based CF approaches for large scale implicit feedback datasets. Instead of ignoring all missing entries in the user-item interaction matrix, they treated all missing entities as negative examples and assigned vastly varying confidence levels to the positive and negative examples. Then, a least square loss function was used to learn the latent features of users and items. To simplify the computation, Pan et al. [19] proposed a sampling framework to collect a fraction of missing entries in the user-item matrix as negative examples. Recently, Rendle et al. [21] presented a Bayesian Personalized Ranking (BPR) approach for recommendation with users’ implicit feedback. This method only assumed that the unobserved items (or negative items) were less interesting than the observed items (or positive items) to an individual user. Its objective was to rank the positive items above the negative items in the positive-negative item pairs of each user.

2.2 Structured Sparsity Learning

For feature learning problems in high-dimensional space, sparsity is a desirable property of feature vectors. The L_1 -norm penalty has been widely used to enforce sparsity to the learned feature vectors. Numerous methods have been proposed to solve the L_1 -regularization regression problems in statistics, machine learning and computer vision. The readers can refer to [6] (Chapter 18) for more discussions.

However, the standard L_1 -norm penalty does not consider any dependency (e.g., the group structure) among inputs. This limits its applicability in many complex application scenarios. In recent years, a lot of structured sparsity learning approaches have been proposed to induce different joint sparsity patterns to related variables, by employing more structured constraints on inputs. Yuan and Lin [30] proposed the group lasso model that assumes disjoint group structures of inputs and enforces sparsity on the pre-defined groups of features. It employed a L_1/L_p mixed norm penalty, with $p > 1$, to achieve group level variable selection. In practice, L_1/L_2

and L_1/L_∞ are the most popular mixed norms. Theoretically, it has also been demonstrated that the L_1/L_2 penalty has the potential to improve the accuracy of estimator [9]. Moreover, the group lasso has been extended to allow more complex structures such as hierarchical groups and overlapping groups [11].

These structured sparsity learning approaches have been applied to different real scenarios. For example, Kim and Xing [13] studied the gene selection problems. They proposed a tree-guided group lasso regression method, which applied group lasso to the groups of output variables defined by a hierarchical clustering tree. In [12], Jenatton et al. presented an extension of sparse PCA based on a structured regularization that encoded high order information of the input data. Their experiments on face recognition task demonstrated the advantages of structured approach over unstructured approaches. In addition, the group lasso models have also been applied in time series data analysis problems [14].

2.3 Personalized Location Recommendation

In recent years, the personalized location recommendation in LBSNs has been widely studied. Prior work provides location recommendation via analyzing users’ historical check-ins in LBSNs. Ye et al. [27] reported the first research work about providing location recommendation services for large scale LBSNs. Two friend-based CF approaches were proposed to predict a user’s preferences on locations, considering the preferences of her social friends. The recent work by Bao et al. [2] generated location recommendations, taking advantage of both the user preference and the opinions from local experts in a new city.

To achieve more accurate recommendations, different approaches have been proposed to exploit the geographical characteristics of check-in data. Ye et al. [26] proposed a unified recommendation framework to incorporate the user preference, geographical influence and social influence into one recommendation process. They designed a probabilistic model to capture the geographical influence from a specific user’s visited locations. Note that this approach is different from our instance-level exploitation that considers the geographical influence from a few nearest geographical neighbors of the target location (see Section 4.1 for more details). Cheng et al. [3] studied users’ multi-center check-in behavior and employed a multi-center gaussian model to capture the geographical influence. Then, a matrix factorization model was fused with geographical and social influence for location recommendation in LBSNs. Levandoski et al. [16] proposed a location-aware recommender system, namely LARS, which employed the location-based ratings to provide recommendation. In [17], Liu et al. introduced a probabilistic recommendation framework that employs the geographical influence on a user’s check-in behavior, the user mobility pattern and the user check-in count data for location recommendation. Recently, Zhang et al. [32] developed a unified geo-social recommendation framework, namely iGSLR, in which a kernel density estimation approach was used to personalize the geographical influence on users’ check-in behaviors as individual distributions. Wang et al. [25] proposed algorithms that generated recommendations based on the past user behavior (visited locations), the longitude and latitude of each location, the social relationships among users, and the similarity between users. Moreover, Yin et al. [28] introduced a location-content-aware recommender system that offered a user a set of venues or events, considering both personal interest and the local preference of each individual city.

Summary. The aforementioned PLRs exploit geographical characteristics from user perspective. Our work differs from existing work in that it exploits geographical characteristics from location

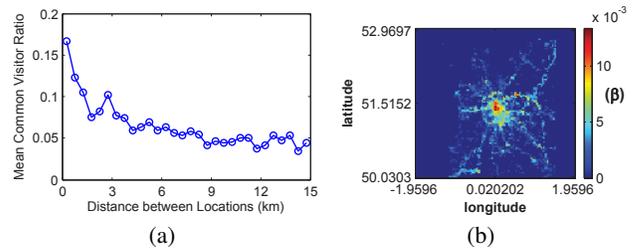


Figure 1: It shows the instance-level geographical neighborhood characteristics: (a) the relationship between common visitor ratio β and the distance between locations, (b) the common visitor ratio between a typical location (id: 23092, latitude: 51.5144313985, longitude: -0.152746439) and other locations.

perspective, with an emphasis on the geographical neighborhood. Moreover, it also proposes a unified framework to consider two levels of geographical neighborhood characteristics for location recommendation.

3. PRELIMINARIES

In this section, we first analyze geographical neighborhood characteristics of the check-in data from location perspective. Then, we briefly introduce matrix factorization for recommendation.

3.1 Empirical Data Analysis

To gain a better understanding of users’ check-in activities, we study the check-in data collected from Gowalla, a popular LBSN (see Section 5.1.1 for details). More specifically, we study the geographical neighborhood characteristics at two levels: instance-level and region-level. We make two observations from the data analysis: a) nearest neighboring locations tend to share more similar user preferences; and b) locations in the same geographical region may share similar user preferences. These patterns are independent of any particular user. We now report the details of the data analysis.

Instance Level. At the instance level, we measure the similarity of users’ check-ins at two different locations ℓ_j and ℓ_k using Jaccard similarity coefficient, denoted by $\beta(\ell_j, \ell_k)$. This measure is also known as the *common visitor ratio* of the two locations in our discussion.

$$\beta(\ell_j, \ell_k) = \frac{|\mathcal{U}(\ell_j) \cap \mathcal{U}(\ell_k)|}{|\mathcal{U}(\ell_j) \cup \mathcal{U}(\ell_k)|}, \quad (1)$$

where $\mathcal{U}(\ell_j)$ and $\mathcal{U}(\ell_k)$ denote the two sets of users who have visited ℓ_j and ℓ_k respectively.

Figure 1(a) plots the relationship between $\beta(\ell_j, \ell_k)$ and the geographical distance $d(\ell_j, \ell_k)$ between ℓ_j and ℓ_k .² As shown in Figure 1(a), β decreases with the increasing of d , which indicates that geographically adjacent locations tend to share more common visitors. In other words, neighboring locations are more likely to be visited by the same set of users, or users have similar preferences on neighboring locations. As an example, Figure 1(b) shows the common visitor ratio between a randomly sampled location (id: 23092, latitude: 51.5144313985, longitude: -0.152746439) and the other locations. It reflects that location 23092 shares more common visitors with its neighboring locations.

Region Level. To study the geographical characteristics of locations at region level, we cluster the locations into groups. The clus-

²Here, the common visitor ratio is averaged over all pairs of locations with same distances, and the distance between two locations is computed by using the Haversine formula.

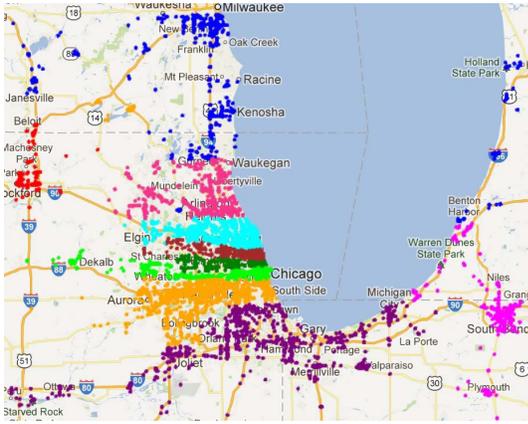


Figure 2: The checked-in locations around Chicago are clustered into 10 groups, denoted in different colors. It shows the region-level geographical neighborhood characteristics of check-in data.

tering can be conducted by using different similarity/distance measures between locations, e.g., geographical distance, user check-in similarity (i.e., $\beta(\ell_j, \ell_k)$), or the combination of the two.³ For illustration purpose, here we present a clustering result by using user check-in similarity with geographical constraint, similar to that reported in [4]. A spectral clustering algorithm is utilized to group the locations [18]. More specifically, we compute an affinity matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ on all observed locations $\{\ell_j\}_{j=1}^n$. For a given location ℓ_j , let $\mathcal{N}(\ell_j)$ denote the set of N closest locations to ℓ_j by the geographical distance. In this analysis, we set N at 10. The element M_{jk} in \mathbf{M} is determined as,

$$M_{jk} = \begin{cases} \beta(\ell_j, \ell_k) + \varepsilon, & \text{if } \ell_j \in \mathcal{N}(\ell_k) \text{ or } \ell_k \in \mathcal{N}(\ell_j) \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where ε is a small constant used to keep each location connecting to arbitrary other locations either directly or indirectly. Then, the spectral algorithm is used to partition locations into groups, considering users' check-ins at each location and the geographical distances between locations.

Figure 2 shows an example of 10 clusters of the locations around Chicago area. The location clusters are denoted in different colors. Observe that the locations are distributed in geographical regions and the shapes of the regions are determined by user check-in behavior. In other words, locations in the same geographical region may share similar user preferences. Suggested in earlier studies, each of these geographical regions is usually associated with a specific function such as education, business, entertainment and so on, considering the category information of the locations in the region [4, 29]. Note that, this observation is consistent with the instance-level geographical characteristics. Nevertheless, a region refers to a much larger geographical area than the area containing a few nearest neighbors of a specific location.

Motivated by the two observations, we propose incorporating the above geographical characteristics of check-in data for location recommendation, so as to improve location recommendation accuracy.

3.2 Recommendation by Matrix Factorization

The recommendation task studied in this paper can be defined as: given the historical check-ins of m users $\{u_i\}_{i=1}^m$ over n locations $\{\ell_j\}_{j=1}^n$, for a target user, recommend her a set of locations

³The three kinds of similarity/distance measures are evaluated in our experiments (see Section 5.2.3).

that she might be interested in but has not visited before. Here, we first introduce a rating matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$ to describe users' preferences on locations, where each element $R_{ij} \in \{0, 1\}$ denotes u_i 's preference on ℓ_j . If u_i has checked-in at ℓ_j at least once, we set R_{ij} to 1, otherwise, we set R_{ij} to 0. Note that $R_{ij} = 0$ does not explicitly indicate u_i is not interested in ℓ_j . It can also be caused by that u_i does not know ℓ_j .

In the literature, matrix factorization based approaches are the most successful and widely used recommendation methods [15]. The primary idea of matrix factorization is to map both users and locations into a shared space with dimension $r \ll \min(m, n)$, and represent u_i and ℓ_j using $\mathbf{U}_i \in \mathbb{R}^{1 \times r}$ and $\mathbf{L}_j \in \mathbb{R}^{1 \times r}$, which are known as the latent factors of u_i and ℓ_j respectively. The preference of u_i on ℓ_j can then be approximated using

$$\hat{R}_{ij} = \mathbf{U}_i \mathbf{L}_j^\top. \quad (3)$$

Biases for users and locations can also be incorporated in Eq. (3) to produce more accurate models [15]. In general, we denote the latent factors of all users and locations by $\mathbf{U} \in \mathbb{R}^{m \times r}$ and $\mathbf{L} \in \mathbb{R}^{n \times r}$ respectively, where \mathbf{U}_i is the i^{th} row in \mathbf{U} and \mathbf{L}_j is the j^{th} row in \mathbf{L} . The model parameters, \mathbf{U} and \mathbf{L} , can be learned by minimizing a weighted regularized square error loss [8, 15, 19],

$$\min_{\mathbf{U}, \mathbf{L}} \frac{1}{2} \sum_{i,j} W_{ij} (R_{ij} - \hat{R}_{ij})^2 + \frac{\lambda_1}{2} \|\mathbf{U}\|_F^2 + \frac{\lambda_2}{2} \|\mathbf{L}\|_F^2, \quad (4)$$

where W_{ij} gives a confidence level to the preference feedback R_{ij} , $\|\cdot\|_F$ is the Frobenius norm of a matrix, λ_1 and λ_2 are regularization parameters.

Once the latent factors \mathbf{U} and \mathbf{L} have been learned, the preference value associated with any user-location pair (u_i, ℓ_j) can be predicted using Eq. (3).

4. EXPLOITATION OF NEIGHBORHOOD CHARACTERISTICS

In this section, we present the details of IReMF, the objective of which is to exploit two levels (i.e., instance-level and region-level) of neighborhood characteristics to provide more accurate location recommendations. Firstly, at the instance level, a geographical weighting strategy is used to consider the strong relations among users' preferences on the target location and a few nearest neighbors of it. Secondly, at the region level, a group lasso penalty is introduced to take advantage of the geographical region structure derived from the check-in data.

4.1 Instance-Level Exploitation

Through mapping users and locations into a shared latent space, MF can effectively estimate the overall relations associated with most or all locations. However, the classical MF based approach introduced in Section 3.2 ignores the strong relations among geographical nearest neighboring locations.

Our empirical analysis of check-in data shows that nearest neighboring locations tend to share more common visitors. Inspired by this observation, we propose to characterize u_i 's preference on ℓ_j using her preferences on a few nearest neighboring locations of ℓ_j . Thus, we modify the prediction of R_{ij} as,

$$\hat{R}_{ij}^{\text{new}} = \alpha \mathbf{U}_i \mathbf{L}_j^\top + (1 - \alpha) \frac{1}{Z(\ell_j)} \sum_{\ell_k \in \mathcal{N}(\ell_j)} \text{Sim}(\ell_j, \ell_k) \mathbf{U}_i \mathbf{L}_k^\top, \quad (5)$$

where $\alpha \in [0, 1]$ is the instance weighting parameter used to control the influence from neighboring locations, $\mathcal{N}(\ell_j)$ denotes the

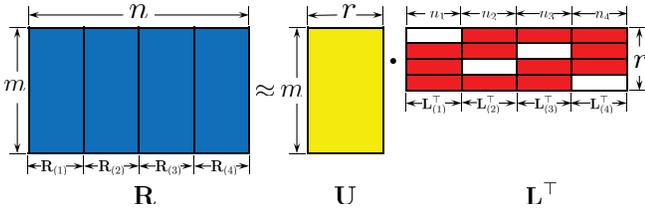


Figure 3: An example of the structured sparsity assumption of the location latent factors. The locations are clustered into 4 groups, and both users and locations are mapped into a shared latent space with dimension 4. In \mathbf{L}^\top , the red and white cells denote nonzero and zeros values respectively.

set of N nearest neighboring locations of ℓ_j . In the following experiments, we empirically set N at 10^4 . In Eq. (5), $Sim(\ell_j, \ell_k)$ is the geographical weight denoting the geographical influence of ℓ_k on ℓ_j . $Z(\ell_j)$ is a normalizing factor formulated as $Z(\ell_j) = \sum_{\ell_k \in \mathcal{N}(\ell_j)} Sim(\ell_j, \ell_k)$. We define $Sim(\ell_j, \ell_k)$ using a Gaussian function as follows based on our observation in Figures 1(a) and 1(b).

$$Sim(\ell_j, \ell_k) = e^{-\frac{\|\mathbf{x}_j - \mathbf{x}_k\|^2}{\sigma^2}} \quad \forall \ell_k \in \mathcal{N}(\ell_j), \quad (6)$$

where \mathbf{x}_j and \mathbf{x}_k represent the geographical coordinates (latitude and longitude) of ℓ_j and ℓ_k respectively, σ is a constant which is empirically set at 0.1 in our experiments.

Note that this instance-level exploitation is essentially different from the approach proposed in [26]. Our approach is from location perspective and considers the relationship between a target location ℓ_j and a few nearest geographical neighbors $\mathcal{N}(\ell_j)$ of ℓ_j . The prediction of u_i 's preference on ℓ_j is determined by the characteristics of u_i , ℓ_j and $\mathcal{N}(\ell_j)$. However, in [26], the probability that u_i would check-in at ℓ_j is determined by the distances between ℓ_j and u_i 's **visited locations** $\mathcal{L}(u_i)$ (i.e., from user perspective). Usually, $\mathcal{N}(\ell_j)$ and $\mathcal{L}(u_i)$ are entirely different or with a very small overlap, as u_i has only visited a small proportion of locations. Therefore, **these two methods are modeling two different kinds of geographical characteristics**. In all experiments, the proposed instance-level exploitation outperforms the approach in [26] (see Section 5.2.1 for details).

4.2 Region-Level Exploitation

As aforementioned, locations in the same geographical region may share similar user preferences. In this section, we are interested in incorporating this region-level characteristics into the learning of latent factors (or latent features) of users and locations.

In the feature learning problems, the structured sparsity learning approaches (e.g., group lasso) have been widely used to consider the prior knowledge of the group structure of inputs. Inspired by its successful applications in real scenarios [10, 12–14], we propose to utilize group lasso to exploit the geographical region structure of locations, via assuming the latent factors of locations from the same region share the same sparsity pattern, namely the structured sparsity assumption of location latent factors. In this assumption, the locations are assumed to be clustered into G disjoint groups: $\{\mathcal{L}_{(g)}\}_{g=1}^G$. The location latent factors \mathbf{L} are divided into G groups as $\mathbf{L} = \{\mathbf{L}_{(g)}\}_{g=1}^G$, where $\mathbf{L}_{(g)} \in \mathbb{R}^{n_g \times r}$ denotes the latent factors of locations belonging to $\mathcal{L}_{(g)}$, n_g is the number of locations in

⁴We have also evaluated the performances with respect to different setting of N . The experimental results indicate that the location recommendation accuracy is not very sensitive to the size of nearest neighbors.

$\mathcal{L}_{(g)}$. Accordingly, the rating matrix \mathbf{R} is divided into G groups as $\mathbf{R} = \{\mathbf{R}_{(g)}\}_{g=1}^G$, where $\mathbf{R}_{(g)} \in \mathbb{R}^{m \times n_g}$.

Figure 3 gives an example of the structured sparsity assumption. As shown in Figure 3, the column vectors in $\mathbf{L}_{(2)}^\top$ share the same sparsity pattern since all their third rows are zeros. Therefore, the reconstruction of elements in $\mathbf{R}_{(2)}$ only relies on the first, second and fourth row components in $\mathbf{L}_{(2)}^\top$. With this sparsity structure of \mathbf{L} , the associations of latent factors of users and locations can be understood at the group level instead of instance level, as the geographical structure information of locations is embedded in the location latent factors.

Taking into account the structured sparsity assumption, we use a group lasso penalty of \mathbf{L} as,

$$\lambda_3 \sum_{g=1}^G \sum_{d=1}^r \omega_g \|\mathbf{L}_{(g)}^d\|_2, \quad (7)$$

to integrate the geographical region structure information into the learning of latent factors \mathbf{U} and \mathbf{L} . In Eq. (7), λ_3 is the group sparsity regularization parameter. $\mathbf{L}_{(g)}^d \in \mathbb{R}^{n_g \times 1}$ is the d^{th} column vector in $\mathbf{L}_{(g)}$. ω_g is a weight assigned to $\mathbf{L}_{(g)}^d$. A simple setting of ω_g is $\omega_g = n_g^{\frac{1}{2}}$, which enables the amount of penalization to be adjusted according to the size of each group [30].

4.3 IReNMF

In this section, we present the final formulation of IReNMF. By plugging Eq. (5) and Eq. (7) into Eq. (4), the optimization problem of IReNMF is formulated as follows,

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{L}} \mathcal{F}(\mathbf{U}, \mathbf{L}) = & \frac{1}{2} \sum_{i,j} W_{ij} (R_{ij} - \hat{R}_{ij}^{new})^2 + \frac{\lambda_1}{2} \|\mathbf{U}\|_F^2 \\ & + \frac{\lambda_2}{2} \|\mathbf{L}\|_F^2 + \lambda_3 \sum_{g=1}^G \sum_{d=1}^r \omega_g \|\mathbf{L}_{(g)}^d\|_2. \end{aligned} \quad (8)$$

For simplicity, we rewrite the problem in Eq. (8) as,

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{L}} \mathcal{F}(\mathbf{U}, \mathbf{L}) = & \frac{1}{2} \|\widetilde{\mathbf{W}} \circ (\mathbf{R} - \mathbf{U}\mathbf{L}^\top \mathbf{S}_\alpha)\|_F^2 + \frac{\lambda_1}{2} \|\mathbf{U}\|_F^2 \\ & + \frac{\lambda_2}{2} \|\mathbf{L}\|_F^2 + \lambda_3 \sum_{g=1}^G \sum_{d=1}^r \omega_g \|\mathbf{L}_{(g)}^d\|_2, \end{aligned} \quad (9)$$

where $\widetilde{\mathbf{W}} \in \mathbb{R}^{m \times n}$ is a weighting matrix. \circ denotes the Hadamard product of two matrices. $\mathbf{S}_\alpha = \alpha \mathbf{I} + (1 - \alpha) \mathbf{S}^\top$. $\mathbf{I} \in \mathbb{R}^{n \times n}$ is an identity matrix. $\mathbf{S} \in \mathbb{R}^{n \times n}$, in which each element $S_{jk} = Sim(\ell_j, \ell_k) / Z(\ell_j)$.

In general, as u_i 's check-in times C_{ij} at ℓ_j grows, we have a strong indication that u_i indeed likes ℓ_j . Therefore, a simple setting of the element \widetilde{W}_{ij} in $\widetilde{\mathbf{W}}$ is as,

$$\widetilde{W}_{ij} = (1 + \delta C_{ij})^{\frac{1}{2}} \quad (10)$$

where the constant δ is used to control the rate of increase. Empirically, we set $\delta = 10$ in our experiments.

4.3.1 Optimization Algorithms

Here, we present the solution to the optimization problem stated in Eq. (9), which is a non-convex problem, and can be further decomposed into two convex optimization problems by fixing one variable (\mathbf{U} or \mathbf{L}) and optimizing the other one. Therefore, we use the following two-step alternating optimization strategy to solve the problem in Eq. (9).

Algorithm 1: APG Optimization

Input : $\mathbf{L}^{(0)}$, λ_2 , λ_3 , $\tau^{(0)}$, ε , η , $t^{(0)}$, $\{\omega_g\}_{g=1}^G$
Output: $\mathbf{L}^{(k)}$

- 1 Initialize: $\mathbf{L}^{(1)} \leftarrow \mathbf{L}^{(0)}$, $k \leftarrow 1$, $\sigma \leftarrow 1.0$
- 2 Use Eq. (12) and $\mathbf{L}^{(0)}$ to calculate $\varphi(\mathbf{L}^{(0)})$
- 3 **while** $k \leq \text{maxIter}$ && $\sigma \geq \varepsilon$ **do**
 - // maxIter is the maximum number of iterations
 - 4 $\tau \leftarrow \eta\tau^{(k-1)}$, $t^{(k)} \leftarrow \frac{1+\sqrt{1+4(t^{(k-1)})^2}}{2}$,
 $\mathbf{X}^{(k)} \leftarrow \mathbf{L}^{(k)} + \frac{t^{(k-1)}-1}{t^{(k)}}(\mathbf{L}^{(k)} - \mathbf{L}^{(k-1)})$
 - 5 **while** 1 **do**
 - 6 $\mathbf{Z} \leftarrow \mathbf{X}^{(k)} - \frac{1}{\tau}\nabla p(\mathbf{X}^{(k)})$
 - 7 Use Eq. (16) to calculate \mathbf{L}^*
 - 8 Use Eq. (13) and Eq. (14) to compute $p(\mathbf{L}^*)$ and $p_\tau(\mathbf{L}^*, \mathbf{X}^{(k)})$;
 - 9 **if** $p(\mathbf{L}^*) \leq p_\tau(\mathbf{L}^*, \mathbf{X}^{(k)})$ **then**
 - 10 $\tau^{(k)} \leftarrow \tau$, **break**;
 - 11 **else**
 - 12 $\tau \leftarrow \frac{\tau}{\eta}$
 - 13 $\mathbf{L}^{(k)} \leftarrow \mathbf{L}^*$
 - 14 Use Eq. (12) to update $\varphi(\mathbf{L}^{(k-1)})$ and $\varphi(\mathbf{L}^{(k)})$
 - 15 $\sigma \leftarrow \frac{|\varphi(\mathbf{L}^{(k)}) - \varphi(\mathbf{L}^{(k-1)})|}{|\varphi(\mathbf{L}^{(k-1)})|}$
 - 16 $k \leftarrow k + 1$
- 17 **Return** $\mathbf{L}^{(k)}$;

Optimize U: when \mathbf{L} is fixed, the optimization problem with respect to \mathbf{U} is as follows,

$$\min_{\mathbf{U}} \phi(\mathbf{U}) = \frac{1}{2} \|\widetilde{\mathbf{W}} \circ (\mathbf{R} - \mathbf{U}\mathbf{L}^\top \mathbf{S}_\alpha)\|_F^2 + \frac{\lambda_1}{2} \|\mathbf{U}\|_F^2. \quad (11)$$

To solve the problem in Eq. (11), we directly use the alternating least squares (ALS) algorithm proposed in [8, 19].

Optimize L: when \mathbf{U} is fixed, the optimization problem becomes,

$$\min_{\mathbf{L}} \varphi(\mathbf{L}) = p(\mathbf{L}) + \frac{\lambda_2}{2} \|\mathbf{L}\|_F^2 + \lambda_3 \sum_{g=1}^G \sum_{d=1}^r \omega_g \|\mathbf{L}_{(g)}^d\|_2, \quad (12)$$

where

$$p(\mathbf{L}) = \frac{1}{2} \|\widetilde{\mathbf{W}} \circ (\mathbf{R} - \mathbf{U}\mathbf{L}^\top \mathbf{S}_\alpha)\|_F^2. \quad (13)$$

The problem in Eq. (12) can be solved using the accelerated proximal gradient (APG) method [24]. The key point of APG approaches is to construct the Taylor approximation of $p(\mathbf{L})$ at the point \mathbf{X} as,

$$p_\tau(\mathbf{L}, \mathbf{X}) = \frac{\tau}{2} \|\mathbf{L} - \mathbf{Z}\|_F^2 - \frac{1}{2\tau} \|\nabla p(\mathbf{X})\|_F^2 + p(\mathbf{X}) \quad (14)$$

where $\tau > 0$ and $\mathbf{Z} = \mathbf{X} - \frac{1}{\tau}\nabla p(\mathbf{X})$. For a fixed point \mathbf{X} , we need to solve the following optimization problem,

$$\min_{\mathbf{L}} \psi(\mathbf{L}) = \frac{\tau}{2} \|\mathbf{L} - \mathbf{Z}\|_F^2 + \frac{\lambda_2}{2} \|\mathbf{L}\|_F^2 + \lambda_3 \sum_{g=1}^G \sum_{d=1}^r \omega_g \|\mathbf{L}_{(g)}^d\|_2. \quad (15)$$

The solution to the problem in Eq. (15) is given in Theorem 1.

THEOREM 1. *Let \mathbf{L}^* be the optimal solution to problem (15). \mathbf{L}^* is unique and can be calculated via a soft-threshold operator*

Algorithm 2: lRenMF Optimization

Input : \mathbf{R} , $\widetilde{\mathbf{W}}$, r , ε , $\{\lambda_i\}_{i=1}^3$, $\{\omega_g\}_{g=1}^G$
Output: $\mathbf{U}^{(k)}$, $\mathbf{L}^{(k)}$

- 1 Initialize $\mathbf{U}^{(0)}$ and $\mathbf{L}^{(0)}$ with random elements, $k \leftarrow 1$,
 $\sigma \leftarrow 1.0$
- 2 **while** $k \leq \text{maxIter}$ && $\sigma \geq \varepsilon$ **do**
 - // maxIter is the maximum number of iterations
 - 3 Compute $\mathbf{U}^{(k)}$ by solving the problem in Eq. (11)
 - 4 Compute $\mathbf{L}^{(k)}$ by solving the problem in Eq. (12)
 - 5 $\sigma \leftarrow \frac{|\mathcal{F}(\mathbf{U}^{(k)}, \mathbf{L}^{(k)}) - \mathcal{F}(\mathbf{U}^{(k-1)}, \mathbf{L}^{(k-1)})|}{|\mathcal{F}(\mathbf{U}^{(k-1)}, \mathbf{L}^{(k-1)})|}$
 - 6 $k \leftarrow k + 1$
- 7 **Return** $\mathbf{U}^{(k)}$ and $\mathbf{L}^{(k)}$;

defined as

$$[\mathbf{L}^*]_{(g)}^d = \begin{cases} \frac{\mathbf{z}_{(g)}^d (\|\mathbf{z}_{(g)}^d\|_2 - \frac{\lambda_3 \omega_g}{\tau})}{(1 + \frac{\lambda_2}{\tau}) \|\mathbf{z}_{(g)}^d\|_2}, & \text{if } \|\mathbf{z}_{(g)}^d\|_2 > \frac{\lambda_3 \omega_g}{\tau} \\ \mathbf{0}, & \text{otherwise,} \end{cases} \quad (16)$$

where $[\mathbf{L}^*]_{(g)}^d \in \mathbb{R}^{n_g \times 1}$ and $\mathbf{z}_{(g)}^d \in \mathbb{R}^{n_g \times 1}$ are the sub-vectors at the d^{th} dimension of the g^{th} group of \mathbf{L}^* and \mathbf{Z} respectively.

The proof of Theorem 1 is shown in the Appendix. The details of the APG solution to Eq. (12) are presented in Algorithm 1.

The entire optimization procedure for the problem in Eq. (9) is summarized in Algorithm 2. Although the objective function of lRenMF (like the objective function of other “standard” recommender systems) is certainly non-convex [15] and subject to local minima, in our experiments it yields stable recommendation accuracy when restarting with different initial conditions.

5. EXPERIMENTS

In this section, we first introduce the experimental settings and then present the analysis of experimental results. The experiments focus on the following aspects: (1) the comparison of the performances of our proposed method and baseline recommendation algorithms, (2) the sensitivity of lRenMF to different parameter settings, and (3) the impact of different location clustering strategies.

5.1 Experimental Settings

5.1.1 Dataset Description

The experimental data used in this study was collected from Gowalla, a popular LBSN, which has more than 600,000 users since November 2010 and was acquired by Facebook in December 2011.⁵ In practice, we used the Gowalla APIs to collect user profiles and check-in data made before June 1, 2011. Finally, we have obtained 36,001,959 check-ins made by 319,063 users over 2,844,076 locations. Each check-in contains user id, location id, longitude, latitude, timestamp, etc.. To evaluate the performances of lRenMF, we construct four datasets via extracting the check-in data generated in four popular cities: Berlin, London, Chicago, and San Francisco. The detailed statistics of the check-in data in the four datasets are summarized in Table 1.

⁵<http://techcrunch.com/2011/12/02/report-facebook-has-acquired-gowalla/>

Table 1: Statistics of datasets.

	Berlin	Chicago	London	S.F.
#users	5,510	13,852	17,112	21,591
#locations	15,528	38,505	63,466	66,142
#check-ins	238,972	490,998	942,877	1,545,407
avg. #users per loc.	23.98	20.28	32.11	41.35
avg. #loc. per user	8.51	7.30	8.66	13.50

5.1.2 Data Partition

In our experiments, each of the four datasets is partitioned into three non-overlapping parts. More specifically, within one dataset, if a user u_i has visited n_i different locations $\mathcal{L}(u_i) = \{\ell_i^1, \ell_i^2, \dots, \ell_i^{n_i}\}$, we randomly choose 60% of her visited locations $\mathcal{L}(u_i)$ as training data, and choose another 10% of $\mathcal{L}(u_i)$ as development data to tune parameters. The remaining 30% of $\mathcal{L}(u_i)$ are used as testing data to evaluate the effectiveness of recommendation methods. Similar experimental settings have been used to validate the performances of location recommendation methods in previous work [3, 26, 31]. After splitting the check-in data, the densities of the training data of Berlin, Chicago, London and San Francisco datasets are 1.17×10^{-3} , 4.13×10^{-4} , 3.84×10^{-4} , and 4.68×10^{-4} respectively.

5.1.3 Evaluation Metrics

In this work, the quality of location recommendations is assessed using two metrics (precision and recall), as suggested by [3, 26]. The precision and recall of the top- K location recommendations to a target user are denoted by $P@K$ and $R@K$ respectively. $P@K$ measures the ratio of recovered locations to the K recommended locations, and $R@K$ measures the ratio of recovered locations to the set of locations in the testing data. Given an individual user u_i , $\mathcal{L}^T(u_i)$ denotes the set of corresponding visited locations in the testing data, and $\mathcal{L}^R(u_i)$ denotes the set of recommended locations by a method. The definitions of $P@K$ and $R@K$ are:

$$\begin{aligned}
 P@K &= \frac{1}{|\mathcal{T}|} \sum_{u_i \in \mathcal{T}} \frac{|\mathcal{L}^T(u_i) \cap \mathcal{L}^R(u_i)|}{K}, \\
 R@K &= \frac{1}{|\mathcal{T}|} \sum_{u_i \in \mathcal{T}} \frac{|\mathcal{L}^T(u_i) \cap \mathcal{L}^R(u_i)|}{|\mathcal{L}^T(u_i)|}, \quad (17)
 \end{aligned}$$

where \mathcal{T} denotes the set of users in the testing data. Specifically, we choose $P@5$, $P@10$, $R@5$, and $R@10$ as evaluation metrics in our experiments.

Note that all the four datasets in this study have low density, which usually leads to relatively low precision and recall. Thus, the low values of precision and recall shown in our experiments are reasonable and in similar range of values reported in other work [3, 26, 31]. In this paper, we present the relative improvements our approach achieved, instead of the absolute values.

5.1.4 Evaluated Recommendation Algorithms

We compare the performances of the following personalized location recommendation methods:

UserCF: This is the user-based CF approach used in [26] that predicts a user’s preferences, considering the preferences of other similar users. The user similarity is computed using cosine similarity and $k = 150$.

ItemCF: This is the item-based CF approach [22] that predicts a user’s preference on a target location, considering her preferences on similar locations. The location similarity is computed using cosine similarity and $k = 150$.

WRMF: This is the weighted regularized matrix factorization model designed for processing large scale implicit feedback datasets [8, 19]. It is a special case of IReNMF without considering the geographical characteristics of check-in data (via setting $\alpha = 1$ and $\lambda_3 = 0$ in Eq. (9)).

BPRMF: The Bayesian personalized ranking (BPR) is another state-of-the-art CF framework for implicit feedback data [21]. BPRMF indicates the choice of using matrix factorization as the learning model in the BPR optimization criterion. In this method, the visited locations are assumed to be more interesting than unvisited locations to an individual user.

GeoCF: This is the user preference/geographical influence based recommendation model proposed in [26]. In this method, a power-law distribution model is used to capture the geographical influence from a user’s visited locations. A unified location recommendation framework is used to linearly combine the geographical influence and the user preference derived by a user-based CF approach.

MGMMF: This approach [3] is based on the observation that a user tends to check-in around several centers. A Multi-center Gaussian Model (MGM) is employed to model the probability that a user will check-in at a given location. Moreover, a fusion framework is used to fuse the user preference (derived by a MF model) and the MGM check-in probability for location recommendation.

InMF: It is a special case of IReNMF that only exploits the instance-level geographical neighborhood characteristics for location recommendation (via setting $\lambda_3 = 0$ in Eq. (9)).

RenMF: It is a special case of IReNMF that only exploits the region-level geographical neighborhood characteristics for location recommendation (via setting $\alpha = 1$ in Eq. (9)).

IReNMF: This is our proposed approach that exploits both instance- and region-level geographical neighborhood characteristics for location recommendation.

In the MF-based approaches, we set the dimension of latent space r at 200, considering both efficiency and effectiveness. The regularization parameters λ_1 and λ_2 of IReNMF, are chosen by cross-validation and finally set at 0.015. The group sparsity regularization parameter λ_3 is set at 1 on all datasets. The instance weighting parameter α of IReNMF is set at 0.4, 0.4, 0.6, 0.4 on Berlin, Chicago, London, and San Francisco datasets respectively. The impact of different settings of α and λ_3 is discussed in Section 5.2.2. As a preprocessing step, we use the k -means algorithm to cluster locations into 50 groups according to their latitudes and longitudes. The impact of the region structures discovered by different clustering strategies is discussed in Section 5.2.3. In the experiments, we ran MF-based approaches five times and report the average results on the top-5 and top-10 recommendations respectively.

5.2 Experimental Results

5.2.1 Performance Comparison

We report two sets of results for methods with and without utilizing geographical characteristics separately.

Methods without Geographical Characteristics. We first compare the performances of four state-of-the-art top- K recommendation algorithms (i.e., UserCF, ItemCF, WRMF, and BPRMF) in the location recommendation task. The precision and recall of these

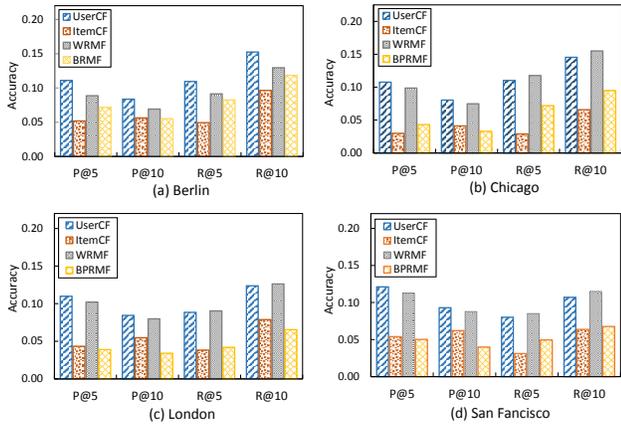


Figure 4: Performances of methods without geographical characteristics.

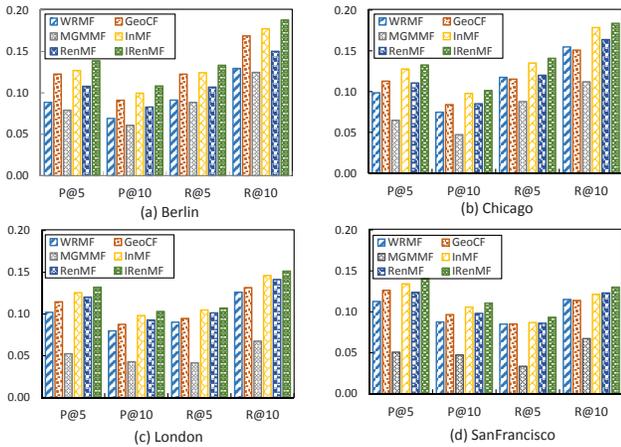


Figure 5: Performances of methods with geographical characteristics.

methods are reported in Figure 4. On all datasets, UserCF outperforms WRMF and BPRMF in terms of precision metrics. Particularly, UserCF attains 12.37% and 132.52% better performance than WRMF and BPRMF over all datasets, in terms of P@5. A possible reason is that the classical MF-based CF approaches only assume a global structure of the check-in data, but ignore the local structure embedded in the check-in data, e.g., the strong relations among nearest geographical neighboring locations. On the other hand, WRMF achieves highest recall values in terms of R@5 and R@10, on the Chicago, London and San Francisco datasets. This indicates that WRMF is an appropriate choice to incorporate geographical characteristics for location recommendation. In addition, as shown in Figure 4, UserCF significantly outperforms ItemCF. One possible explanation is that the user similarity is more accurate than location similarity, as many locations only have a few user check-ins. In this sense, our results are consistent with the results reported in [26].

Methods with Geographical Characteristics. We evaluate the effectiveness of the proposed methods utilizing geographical characteristics from location perspective with other two state-of-the-art methods utilizing geographical characteristics from user perspective [3, 26]. The precision and recall are reported in Figure 5.

Observing from Figure 5, InMF, RenMF, and IRenMF achieve better accuracy than WRMF in terms of both precision and recall, on all datasets. In terms of P@5, InMF outperforms WRMF

by 43.18%, 37.96%, 22.90%, and 19.08% on the four datasets, respectively. RenMF outperforms WRMF by 21.76%, 19.39%, 17.71%, and 9.94% on the four datasets. This indicates that both instance-level and region-level neighborhood characteristics benefit the location recommendation for higher accuracy. In addition, the observation that InMF has better accuracy than RenMF shows that the instance-level characteristics are more effective than the region-level characteristics. This result suggests that nearest neighbouring locations tend to share more user preference. By incorporating both kinds of neighborhood characteristics into the learning of latent factors, the recommendation accuracy of the classical MF model (i.e., WRMF) can be significantly improved. For example, in terms of P@5, the improvements of IRenMF over WRMF are 56.48%, 43.36%, 29.35%, and 24.67% respectively, on all datasets.

To predict a user’s preference on a target location, GeoCF considers the geographical influence from her visited locations (i.e., the geographical characteristics from user’s point of view). In contrast, InMF exploits geographical influence of the nearest neighbouring locations to the target location (i.e., the instance-level geographical characteristics from location’s point of view). Compared to GeoCF, InMF always achieves better results in terms of all measures. For example, in terms of P@10, InMF outperforms GeoCF by 11.78%, on average. This result indicates that the instance-level geographical neighborhood characteristics derived from location perspective have more impacts on the location recommendation accuracy than the geographical characteristics derived from user perspective. Taking advantage of both levels of neighborhood characteristics, IRenMF outperforms GeoCF by 17.90%, in terms of P@10, on average.

Compared to MGMMF, the proposed InMF, RenMF and IRenMF perform significantly better on all datasets. The average improvements of InMF, RenMF and IRenMF over MGMMF, in terms of P@5, are 115.47%, 91.38%, and 127.38%, respectively. Moreover, we also observe that MGMMF performs poorer than WRMF. One possible reason is that MGMMF is based on the probabilistic factor model (PFM), which only models users’ check-in times at locations, instead of users’ preferences on locations.

In Figure 5, IRenMF always achieves the best results in terms of all the evaluation metrics, on all datasets. This demonstrates the advantage of combining both levels of geographical neighborhood characteristics for location recommendation. Most importantly, on average, the proposed IRenMF consistently outperforms the competitors WRMF, GeoCF, and MGMMF, in terms of P@10, by 38.63%, 17.90%, and 116.49% respectively. In all experiments, the standard deviations of the recommendation accuracy of IRenMF are below 0.005, showing the good stability of our method.

5.2.2 Parameter Tuning

In IRenMF, the impact of geographical neighborhood characteristics on location recommendation are controlled by the instance weighting parameter α and group sparsity regularization parameter λ_3 . Figure 6(a) shows P@5 and R@5 of IRenMF (setting λ_3 at 0) with respect to different α ranging from 0 to 1 with an increment of 0.1. As shown in Figure 6(a), IRenMF achieves comparable results on Berlin, Chicago and San Francisco datasets with setting α at 0 and 1. This demonstrates that it is possible to characterize users’ preferences on a target location, using their preferences on a few nearest neighboring locations of it. Specifically, in terms of P@5, we observe that α set at 0.4 on Berlin dataset obtains the best results, as well as 0.4 on Chicago dataset, 0.6 on London dataset, and 0.4 on San Francisco dataset respectively. It suggests that α well tradeoff the importance between a user’s preference on the target

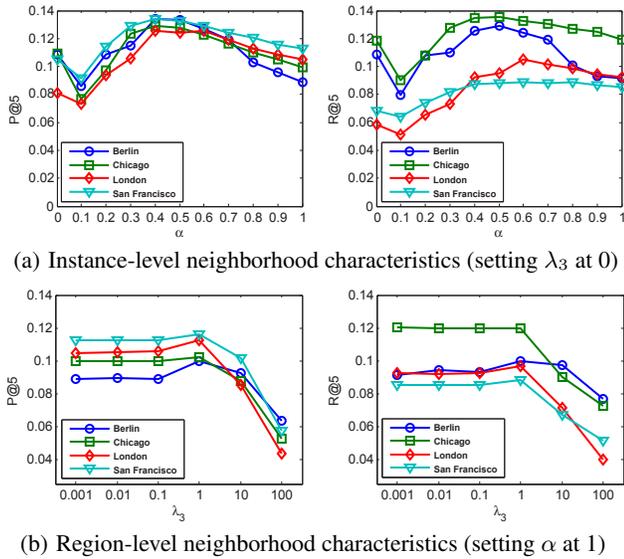


Figure 6: Impact of geographical neighborhood characteristics in IRenMF.

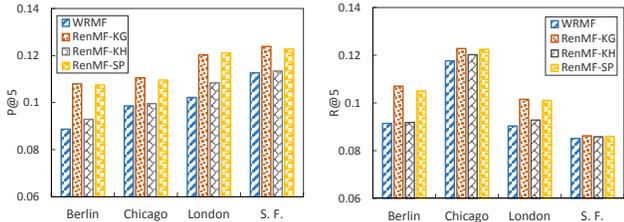


Figure 7: The location recommendation accuracy with respect to different clustering strategies.

location and her preferences on nearest neighboring locations. In particular, we notice that only considering one of them (i.e., $\alpha = 0$ or $\alpha = 1$) will lead to degraded recommendation accuracy.

We test the performance of IRenMF in the grid $\lambda_3 \in \{0.001, 0.01, 0.1, 1, 10, 100\}$, while setting α at 1. The performances of IRenMF are shown in Figure 6(b), which indicates the recommendation accuracy can be gradually influenced by the setting of the group sparsity regularization parameter λ_3 , and $\lambda_3 = 1$ is the most suitable setting in all four datasets. As shown in Figure 6(b), the performance of IRenMF drops drastically by changing λ_3 from 10 to 100. We speculate this is because that the learned latent factors of locations are too sparse, and thus the reconstruction of users' preferences on locations are insufficient to provide accurate location recommendations. For example, in Berlin dataset, the sparsity of the location latent factors is 90.70% and 98.89% when setting λ_3 at 10 and 100 respectively.

5.2.3 Impact of Different Clustering Strategies

In this section, we explore the impact of different clustering strategies on the location recommendation accuracy. To single out the impact of region-level neighborhood characteristics, we simply set the instance weighting parameter α in IRenMF at 1, effectively only evaluating the performance of RenMF.

We study the performance of RenMF with respect to different region structures obtained by clustering locations into 50 groups with the following clustering methods: (1) **KG**, the k -means clustering solely based on the geographical distance between locations; (2) **KH**, the k -means clustering solely based on users' check-ins gen-

erated at each location; (3) **SP**, the spectral clustering [18], which considers both users' check-ins at locations and the geographical distance between locations. Figure 7 presents the recommendation accuracy of RenMF with respect to different region structures obtained by KG, KH and SP. We observe that RenMF-KG, RenMF-KH and RenMF-SP usually outperform WRMF in terms of P@5 and R@5. The average improvements of RenMF-KG, RenMF-KH, and RenMF-SP over WRMF are 17.2%, 4.73%, and 16.74% respectively, in terms of P@5. This demonstrates the importance of region-level neighborhood characteristics to location recommendation. Moreover, RenMF-KG and RenMF-SP achieve better performance than RenMF-KH on all datasets. This indicates that the region structure obtained from the geographical information (longitude and latitude) contributes more to location recommendation accuracy than the interactions between users and locations.

This again highlights the importance of geographical neighborhood characteristics at region level.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel approach, namely Instance-Region Neighborhood Matrix Factorization (IRenMF), to location recommendation by exploiting two levels of geographical neighborhood characteristics from location perspective. By incorporating these two levels of neighborhood characteristics into the learning of latent factors of users and locations, IRenMF has a more accurate modelling of users' preferences on locations. To solve the optimization problem of IRenMF, we propose a novel alternating optimization algorithm, with which IRenMF achieves stable recommendation accuracy. The experiments on real data show that IRenMF leads to significant improvements on the classical MF-based approach, i.e., WRMF, and other state-of-the-art location recommendation models, i.e., GeoCF and MGMMF. On average, IRenMF outperforms WRMF, GeoCF, and MGMMF by 38.63%, 17.90%, and 116.49% respectively, in terms of P@10.

The future work will focus on the following potential directions. First, we would like to evaluate more similarity measures between locations to capture the instance-level neighborhood characteristics. Second, we are also interested in extending IRenMF to exploit other complicated region structures (e.g., hierarchical structures) for more accurate recommendations. Last but not least, we will apply IRenMF to other problems, e.g., business rating prediction [7].

7. REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *TKDE*, 17(6):734–749, 2005.
- [2] J. Bao, Y. Zheng, and M. F. Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *ACM SIGSPATIAL GIS*, 2012.
- [3] C. Cheng, H. Yang, I. King, and M. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. *AAAI*, 2012.
- [4] J. Cranshaw, R. Schwartz, J. Hong, and N. Sadeh. The livelihoods project: Utilizing social media to understand the dynamics of a city. *ICWSM*, 2012.
- [5] A. Das, D. Mayur, and A. Garg. Google News Personalization : Scalable Online Collaborative Filtering. In *WWW*, 2007.
- [6] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. The elements of statistical learning: Data mining, inference and prediction. 2009.

[7] L. Hu, A. Sun, and Y. Liu. Your neighbors affect your ratings: on geographical neighborhood influence to rating prediction. In *SIGIR*, pages 345–354. ACM, 2014.

[8] Y. Hu, F. Park, Y. Koren, and C. Volinsky. Collaborative Filtering for Implicit Feedback Datasets. In *ICDM*, 2008.

[9] J. Huang and T. Zhang. The benefit of group sparsity. *The Annals of Statistics*, 38(4):1978–2004, 2010.

[10] J. Huang, T. Zhang, and D. Metaxas. Learning with structured sparsity. *ICML*, 2009.

[11] L. Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlap and graph lasso. In *ICML*, 2009.

[12] R. Jenatton, G. Obozinski, and F. Bach. Structured sparse principal component analysis. In *AISTATS*, 2010.

[13] S. Kim and E. P. Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *ICML*, pages 543–550, 2010.

[14] M. Kolar, L. Song, and E. P. Xing. Sparsistent learning of varying-coefficient models with structural changes. In *NIPS*, pages 1006–1014, 2009.

[15] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[16] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel. Lars: A location-aware recommender system. In *ICDE*, pages 450–461. IEEE, 2012.

[17] B. Liu, Y. Fu, Z. Yao, and H. Xiong. Learning geographical preferences for point-of-interest recommendation. In *KDD*, pages 1043–1051. ACM, 2013.

[18] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. *NIPS*, 2:849–856, 2002.

[19] R. Pan, Y. Zhou, B. Cao, and N. N. Liu. One-Class Collaborative Filtering. In *ICDM*, 2008.

[20] B. Pradel, S. Sean, and N. Usunier. A case study in a recommender system based on purchase data categories and subject descriptors. In *KDD*, pages 377–385, 2011.

[21] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-thieme. BPR : Bayesian Personalized Ranking from Implicit Feedback. In *UAI*, pages 452–461, 2009.

[22] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295. ACM, 2001.

[23] S. Scellato, A. Noulas, and C. Mascolo. Exploiting place features in link prediction on location-based social networks. In *KDD*, pages 1046–1054, 2011.

[24] K.-C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific J. Optimization*, pages 615–640, 2010.

[25] H. Wang, M. Terrovitis, and N. Mamoulis. Location recommendation in location-based social networks using user check-in data. In *ACM SIGSPATIAL GIS*, pages 364–373. ACM, 2013.

[26] M. Ye, P. Yin, and W. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, pages 325–334, 2011.

[27] M. Ye, P. Yin, and W.-C. Lee. Location Recommendation for Location-based Social Networks. In *Proceedings of the 18th ACM SIGSPATIAL GIS*, pages 458–461, 2010.

[28] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen. Lcars: A location-content-aware recommender system. In *KDD*, pages 221–229. ACM, 2013.

[29] J. Yuan, Y. Zheng, and X. Xie. Discovering regions of different functions in a city using human mobility and pois. In *KDD*, pages 186–194. ACM, 2012.

[30] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[31] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Time-aware point-of-interest recommendation. In *SIGIR*, pages 363–372. ACM, 2013.

[32] J.-D. Zhang and C.-Y. Chow. igsrl: personalized geo-social location recommendation: a kernel density estimation approach. In *ACM SIGSPATIAL GIS*. ACM, 2013.

[33] V. W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang. Collaborative Filtering Meets Mobile Recommendation : A User-centered Approach. In *AAAI*, 2010.

[34] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. *WWW*, pages 1029–1038, 2010.

[35] Y. Zheng. Tutorial on location-based social networks. *WWW*, 2012.

[36] Y. Zheng and X. Xie. Location-based social networks: Locations. *Computing with Spatial Trajectories*, pages 277–308, 2011.

APPENDIX

A. PROOF OF THEOREM 1

Take the sub-gradient of $\psi(\mathbf{L})$ over $\mathbf{L}_{(g)}^d$:

$$\frac{\partial \psi(\mathbf{L})}{\partial \mathbf{L}_{(g)}^d} = \tau(\mathbf{L}_{(g)}^d - \mathbf{Z}_{(g)}^d) + \lambda_2 \mathbf{L}_{(g)}^d + \lambda_3 \omega_g \frac{\partial \|\mathbf{L}_{(g)}^d\|_2}{\partial \mathbf{L}_{(g)}^d}, \quad (18)$$

where $\frac{\partial \|\mathbf{L}_{(g)}^d\|_2}{\partial \mathbf{L}_{(g)}^d} = \frac{\mathbf{L}_{(g)}^d}{\|\mathbf{L}_{(g)}^d\|_2}$, if $\mathbf{L}_{(g)}^d \neq \mathbf{0}$, and $\frac{\partial \|\mathbf{L}_{(g)}^d\|_2}{\partial \mathbf{L}_{(g)}^d} = \{\alpha \in \mathbb{R}^{n_{(g)} \times 1} \mid \|\alpha\|_2 \leq 1\}$, otherwise. When $\mathbf{L}_{(g)}^d \neq \mathbf{0}$, plugging $\frac{\partial \|\mathbf{L}_{(g)}^d\|_2}{\partial \mathbf{L}_{(g)}^d} = \frac{\mathbf{L}_{(g)}^d}{\|\mathbf{L}_{(g)}^d\|_2}$ into Eq. (18), we have

$$(\tau + \lambda_2 + \frac{\lambda_3 \omega_g}{\|\mathbf{L}_{(g)}^d\|_2}) \mathbf{L}_{(g)}^d = \tau \mathbf{Z}_{(g)}^d. \quad (19)$$

Taking the vector ℓ_2 norm on both sides of Eq. (19):

$$\|\mathbf{L}_{(g)}^d\|_2 = \frac{\tau}{\tau + \lambda_2} \|\mathbf{Z}_{(g)}^d\|_2 - \frac{\lambda_3 \omega_g}{\tau + \lambda_2} \quad (20)$$

Since $\mathbf{L}_{(g)}^d \neq \mathbf{0}$, the ℓ_2 norm of $\mathbf{L}_{(g)}^d$ should be a positive real value, which requires $\|\mathbf{Z}_{(g)}^d\|_2 > \frac{\lambda_3 \omega_g}{\tau}$. If the condition $\|\mathbf{Z}_{(g)}^d\|_2 > \frac{\lambda_3 \omega_g}{\tau}$ holds, by plugging Eq. (20) into Eq. (19), we can get

$$\mathbf{L}_{(g)}^d = \frac{\mathbf{Z}_{(g)}^d (\|\mathbf{Z}_{(g)}^d\|_2 - \frac{\lambda_3 \omega_g}{\tau})}{(1 + \frac{\lambda_2}{\tau}) \|\mathbf{Z}_{(g)}^d\|_2}. \quad (21)$$

If $\|\mathbf{Z}_{(g)}^d\|_2 \leq \frac{\lambda_3 \omega_g}{\tau}$, plugging $\mathbf{L}_{(g)}^d = \mathbf{0}$ into Eq. (18), we obtain

$$\frac{\partial \|\mathbf{L}_{(g)}^d\|_2}{\partial \mathbf{L}_{(g)}^d} = \frac{\tau}{\lambda_3 \omega_g} \mathbf{Z}_{(g)}^d. \quad (22)$$

As $\frac{\tau}{\lambda_3 \omega_g} \mathbf{Z}_{(g)}^d \leq 1$, $\frac{\tau}{\lambda_3 \omega_g} \mathbf{Z}_{(g)}^d$ is a valid sub-gradient of $\|\mathbf{L}_{(g)}^d\|_2$ over $\mathbf{L}_{(g)}^d$, when $\mathbf{L}_{(g)}^d = \mathbf{0}$. \square