GeoMF++: Scalable Location Recommendation via Joint Geographical Modeling and Matrix Factorization

DEFU LIAN and KAI ZHENG, University of Electronic Science and Technology of China
YONG GE, University of Arizona
LONGBING CAO, University of Technology Sydney
ENHONG CHEN, University of Science and Technology of China
XING XIE, Microsoft Research

Location recommendation is an important means to help people discover attractive locations. However, extreme sparsity of user-location matrices leads to a severe challenge, so it is necessary to take implicit feedback characteristics of user mobility data into account and leverage location's spatial information. To this end, based on previously developed GeoMF, we propose a scalable and flexible framework, dubbed GeoMF++, for joint geographical modeling and implicit feedback based matrix factorization. We then develop an efficient optimization algorithm for parameter learning, which scales linearly with data size and the total number of neighbor grids of all locations. GeoMF++ can be well explained from two perspectives. First, it subsumes two-dimensional kernel density estimation so that it captures spatial clustering phenomenon in user mobility data; Second, it is strongly connected with widely-used neighbor additive models and graph Laplacian regularized models. We finally evaluate GeoMF++ on two large-scale LBSN datasets with respect to both warm-start and cold-start scenarios. The experimental results show that GeoMF++ consistently outperforms the state-of-the-arts and other competing baselines on both datasets in terms of NDCG and Recall. More importantly, GeoMF++ is much more efficient and scalable with the increase of data size and the dimension of latent space.

CCS Concepts: • **Information systems** → **Location based services**; **Collaborative filtering**; **Personalization**; **Recommender systems**;

Additional Key Words and Phrases: Location Recommendation; Geographical Modeling; LBSNs

ACM Reference Format:
Defu Lian, Kai Zheng, Yong Ge, Longbing Cao, Enhong Chen, and Xing Xie. 2017. GeoMF++: Scalable Location Recommendation via Joint Geographical Modeling and Matrix Factorization. *ACM Transactions on Information Systems* 1, 1, Article 1 (August 2017), 28 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

A preliminary version [16] of this article appeared in Proceedings of SIGKDD 2014.
Author's address: D. Lian and K. Zheng (corresponding author), Big Data Research Center and School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China 611731; emails:{dove.ustc,kevinzheng1983}@gmail.com; Y. Ge, Management Information Systems, University of Arizona, Tucson, USA 85721; email:yongge@email.arizona.edu; L. Cao, Advanced Analytics Institute, University of Technology Sydney, NSW, Australia 2007; email:longbing.cao@uts.edu.au; E. Chen, School of Computer Science and Technology, University of Science and Technology of China, Hefei, China 230027; email:cheneh@ustc.edu.cn; X. Xie, Microsoft Research, Beijing, China 100080; email: xing.xie@microsoft.com.

1046-8188/2017/8-ART1 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

With the popularity of smart mobile devices and the fusion of multiple positioning technologies, it has become much easier for people to acquire real-time information regarding their locations. This development has triggered the advent of location-based social networks (LBSNs), such as Foursquare, Jiepang, Facebook Place and so on. This emergence has not only caused location-based socializing to become a new form of social interaction, but has also helped people speed up familiarization of the surroundings. To achieve the latter goal, location recommendation has become one important means.

Location recommendation has been widely studied recently due to easy access of large-scale user mobility data and inclusion of social network information. User mobility data from the LBSNs (i.e., check-ins) only include the histories of locations where users have been and therefore they likely prefer. The visit frequencies reflect the confidence of her positive preference. Nevertheless, locations where a user has never visited are either really unattractive or undiscovered but potentially appealing. These two cases are usually difficult to differentiate from each other if no auxiliary information provided. These characteristics of mobility behavior lets us consider them implicit feedback and exploit weighted regularized matrix factorization for location recommendation, since it deals with sparsity issues better and outperforms other approaches empirically [11, 25].

However, due to extreme sparsity of user-location matrices, this algorithm still faces severe challenges and suffers from comparatively low performance of recommendation. Fortunately, these challenges can be further alleviated by taking locations' geographical information into account. With the existence of locations' geographical information, spatial clustering phenomenon [29], which indicates individual visited locations tend to cluster together, has been revealed in user mobility behavior on the LBSNs [33]. This phenomenon has been leveraged for location recommendation via geographical modeling. Previous geographical modeling algorithms include parametric [33] and non-parametric [35] approaches for modeling the distribution of distance between any two visited locations. For improving their efficiency of geographical modeling, two-dimensional geo-clustering algorithms over individual visited locations have been proposed [3, 18]. For the sake of avoiding setting the number of clusters, two-dimensional kernel density estimation has been developed [15]. Nevertheless, these geographical modeling algorithms are independent of collaborative filtering, so that they are usually integrated together for location recommendation based on some heuristic approaches, like linear combination.

To this end, we extend two-dimensional kernel density estimation and propose an optimization-based geographical modeling algorithm. It is based on user activity areas and location influential areas, and estimates users' geographical preference for locations by an inner product operator. Moreover, a consistent objective goal with weighted regularized matrix factorization is exploited. Therefore, geographical modeling can be seamlessly incorporated into matrix factorization. In particular, we propose GeoMF to augment latent factors of users and locations with user activity areas and location influential areas, respectively, as shown in Figure 2. In this way, a user's preference for a location is modeled as inner product between them in the augmented space, including the user's both interest-based preference and geographical preference for the location. If the user's geographical preference for the location is non-zero, the activity areas of the user intersect with the influential areas of the location so that this location is reachable from the activity areas of the user. We then propose alternating optimization for learning user/location latent factor with weighted least squares and learning user activity area with sparse and non-negative weighted least squares. However, based on the analysis of time complexity, it suffers from computational issues in particular when learning non-negative user activity areas, though we reveal two important properties about improving efficiency of learning algorithms.

For the sake of improving efficiency and flexibility, we further propose GeoMF++, by mapping location influential areas into the same latent space as that formed by weighted regularized matrix factorization, as shown in Fig 3. Hence, we express a user's geographical preference for a location by dot product of the user's latent factor with the weighted sum of latent factors of the location's influential areas. Different from GeoMF, user activity area is not used in GeoMF++ any more, but can be recovered from user latent factors using non-negative sparse coding, since user latent factors encode propagated geographical influence from location influential areas. We then develop an alternating optimization algorithm for learning user/item/area latent factors, and show that it scales linearly with the data size and the total number of influential areas of all locations. Interestingly, supported by theoretical results about approximating locations' spatial similarity matrix with location influential areas, GeoMF++ is strongly connected with widely-used neighbor additive models and graph Laplacian regularized models.

Finally, we evaluate the proposed algorithms on two large-scale LBSN datasets with respect to both warm-start and cold-start scenarios. The experimental results show that the proposed algorithms benefit a lot from geographical modeling and consistently outperforms several competing baselines on both datasets with respect to both scenarios in terms of NDCG and Recall. We then investigate their training efficiency, and find that GeoMF++ is much more efficient and scalable than the baselines with the increase of data size and the dimension of latent space.

This paper is an extension of our previous paper [16], in which we proposed GeoMF for joint geographical modeling and implicit feedback based matrix factorization, so that geographical modeling can be seamlessly incorporated into matrix factorization. In this article, we further make the following contributions for location recommendation:

- For the sake of improving efficiency and flexibility of GeoMF, we propose GeoMF++ to map location influential areas into the same latent space as that formed by weighted regularized matrix factorization. Based on a newly-developed alternating optimization algorithm, GeoMF++ scales linearly with the data size and with the total number of the influential areas of locations.
- We propose leveraging non-negative sparse coding to recover user activity areas from user latent factors learned from GeoMF++, since they are not parts of GeoMF++ any more. The case studies show that recovered user activity areas are meaningful and reasonable, compared to that learned from GeoMF.
- We provide theoretical results for approximating locations' spatial similarity matrix with location influential areas, and establish strong connection of GeoMF++ with another two widely-used algorithms for joint geographical modeling and matrix factorization.
- We reveal two important properties of GeoMF for improving efficiency of learning non-negative user activity areas, so that it can accelerate the original process of leaning user activity areas.
- We extensively evaluate the proposed algorithms on LBSN datasets with respect to both warm-start and cold-start scenarios. In addition to the self-crawled Jiepang dataset used in our previous work, we also use another large-scale public Gowalla dataset

with 6.4M check-ins from 107K users. The experimental results show that the newly proposed GeoMF++ is consistently superior to GeoMF and other competing baselines on both datasets, in terms of not only training efficiency but also recommendation performance with respect to NDCG and Recall.

2 RELATED WORK

Location recommendation has been an important topic in location-based services. For example, some research has focused on recommending some specific types of locations. Park et al. [26] designed a system based on Bayesian learning with both users' preference and location contexts to recommend restaurants. Similarly, Horozov et al. [10] developed a user-based collaborative filtering system to recommend restaurants to a user, by finding which restaurants similar users have visited before. Zheng et al. [38] designed a random walk style model to do tourism hot spot recommendation by taking into account both users' travel experiences and location attractiveness. In addition to single-type location recommendation mentioned above, there is also some other work considering multiple-activity-type location recommendation. For example, Zheng et al. considered location recommendation and activity recommendation together, so that they can provide location recommendation with respect to different types of activities [37]. The proposed model formulates a location-activity matrix for collaborative filtering and uses some additional information such as location features to help recommendation. Furthermore, a personalized extension for their model was proposed in [36]. The personalized model models a user-location-activity tensor with the GPS data from all users, and employs collective tensor and matrix factorization to do recommendation. Takeuchi and Sugimoto [28] developed an item-based collaborative filtering system to recommend shops to a user, if they are similar to her previously visited shops.

With the growing popularity of location-based social networks, location recommendation is drawing plenty of attention once again. The reasons are two-fold: first, it is possible to obtain large-scale user mobility data; second, several new challenges, including an extremely sparse user-location matrix and the presence of social networks, have arisen from this data. To address these challenges, several methods have been proposed. For example, Ye et al. [33] discovered spatial clustering phenomenon of individual visited locations and characterized it by a power law distributed distance of any pair of visited locations [33]. In addition, they also exploited the similarity between users based on location history and social relationships on social networks for collaborative filtering. To better incorporate social relationships from social networks into collaborative filtering, Noulas et al. [24] conducted random walk with a restart on user-location bipartite graph and social graph simultaneously. With regard to modeling the spatial clustering phenomenon, instead of making the power law distribution assumption, Zhang et al. [35] suggested using kernel density estimation to estimate the distribution of distance between pairs of locations. Concentrating on modeling the distance distribution may ignore the multi-center characteristics of individual visiting locations according to [3]. Thus, the authors tried to apply clustering techniques on individual visited locations for capturing the spatial clustering phenomenon. They also exploited Bayesian non-negative matrix factorization for location recommendation, placing a Gamma prior on non-negative latent factors since this model can capture the skewness of the visit frequency to locations. These two models are then multiplied together since both of them are modeled in a probabilistic way. To improve the ad hoc integration between them, Liu et al. [18] proposed a geographical probabilistic factor analysis framework to takes geo-clustering and Bayesian non-negative matrix factorization into consideration by defining a user's preference for locations as a multiplication of her interest in the locations, the locations' popularity and the distance between her and locations.

In addition to studying the effect of social network information and of spatial clustering phenomenon, there has also been research into studying the impact of context information, e.g., time, and the textual content of locations on location recommendation. For example, in [19, 31, 34], the authors tried to leverage content information of locations via topic modeling to assist location recommendation; In [7], Gao et al. proposed distinguishing a user's latent factors at different times and exploiting several strategies to aggregate a user's time-dependent latent factors; In [4, 20], the authors leveraged the information from previous locations, including the locations themselves, categories and so on, for next location recommendation. The representative algorithms for different kinds of location recommendation have been surveyed and empirically compared with each other in [21], and GeoMF has been recognized as one state-of-the-art algorithm.

Comparing this work with these existing ones, major differences lie in the following perspectives. First, we leverage weighted regularized matrix factorization for location recommendation since according to our experimental results it may be more appropriate than other methods for collaborative filtering from implicit feedback. Second, our model subsumes two-dimensional kernel density estimation and doesn't make any assumption about the distribution of visited locations. Finally, geographical modeling is seamlessly and efficiently incorporated into weighted regularized matrix factorization, and this incorporation explains why modeling the spatial clustering phenomenon helps to deal with the challenge of matrix sparsity. However, we don't take the content information into consideration in our proposed model since we don't have any other information except the categories of locations in our dataset. This information can easily be incorporated into the current framework. Actually, we have tried to incorporate the categories of locations, but haven't see significant improvement. We elaborate it in the discussion of Experiments section.

3 PRELIMINARY

Given users' mobility data, location recommendations operates on a user-location preference matrix $\mathbf{R} = [r_{u,i}] \in \{0,1\}^{M \times N}$, where there are M users, denoted by $\mathcal{U} = \{a_1, \cdots, a_M\}$, and N locations, denoted by $\mathcal{L} = \{l_1, \cdots, l_N\}$. Each entry $r_{u,i}$ indicates whether a user a_u has visited a location l_i. The column vector $\mathbf{r}_u$ corresponds to the u-th row of the matrix $\mathbf{R}$; The column vector $\mathbf{r}^i$ corresponds to the i-th column of the matrix $\mathbf{R}$. The visit frequency matrix $\mathbf{C} \in \mathbb{N}^{M \times N}$ is of the same size as $\mathbf{R}$. $\mathcal{L}_u = \{l_i \in \mathcal{L} | r_{u,i} > 0\}$ denotes all visited locations of the user a_u. Here, upper case bold letters denote matrices, lower case bold letters denote column vectors, and non-bold letters represent scalars.

3.1 Collaborative Filtering for Implicit Feedback

Given the preference matrix $\mathbf{R}$, only users' positive preferences are observed since unvisited locations are either negatively preferred or unknown to users. The visit frequency indicates the confidence level of positive preferences so that a higher visit frequency corresponds to a larger confidence of positive preference. Hence, given the preference matrix $\mathbf{R}$, location recommendation is the One Class Collaborative Filtering (OCCF) problem [11, 25]. In this case, we can randomly sample some negative locations for each user and assign them smaller weights than positive ones. For better dealing with the sparsity challenge, we can even treat all unvisited locations as negative, but at the same time of assigning them smaller weights, we should guarantee the special structure of the weighting matrix for the sake of efficiency. For example, the weighting matrix could follow a sparse and one-rank structure, that is, each entry $w_{u,i}$ of the weighting matrix $\mathbf{W} = [w_{u,i}]$,

$$w_{u,i} = \begin{cases} \alpha(c_{u,i}) + 1 & \text{if } c_{u,i} > 0 \\ 1 & \text{otherwise} \end{cases}, \tag{1}$$

where $\alpha(c_{u,i}) > 0$ is a monotonically increasing function with respect to $c_{u,i}$. In this way, it exactly encodes the observation that the frequency is a confidence of users' positive preference. Based on the weighting matrix, the objective function of collaborative filtering for implicit feedback, called Weighted Regularized Matrix Factorization (WRMF), is represented as follows:

$$\min_{\mathbf{P},\mathbf{Q}} \|\mathbf{W}^{\circ\frac{1}{2}} \odot (\mathbf{R} - \mathbf{P}\mathbf{Q}^T)\|_F^2 + \gamma(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2), \tag{2}$$

where $\odot$ is the Hadamard product operator, i.e., element-wise multiplication of matrices and $\mathbf{W}^{\circ\frac{1}{2}} = [w_{u,i}^{\frac{1}{2}}]$ is Hadamard square root of $\mathbf{W}$. $\|\cdot\|_F$ is the Frobenius norm of matrices, simply a square root of the sum of squared values in matrices. This objective function involves mapping users and locations into a joint latent space with dimension $K \ll \min(M, N)$ via a mapping matrix $\mathbf{P} \in \mathbb{R}^{M\times K}$ and a mapping matrix $\mathbf{Q} \in \mathbb{R}^{N\times K}$ respectively. In the joint latent space, a user's preference for a location is modeled as an inner product between them.

It is worth noting that the approximation error is summed over all entries in the user-location preference matrix, but it can be efficiently reduced via alternating least squares and its time complexity in each iteration is still in proportion to the total number of non-zero entries in the user-location preference matrix. We will provide detailed analysis in subsequent sections.

4 GEOMF

Weighted regularized matrix factorization has shown its superiority in most implicit feedback datasets. However, due to the inclusion of geographical information of locations, there is still room for improvement to this algorithm. Although some recent studies have leveraged spatial clustering phenomenon for improving location recommendation [3, 7, 18, 35], most of them are almost independent of the procedure for collaborative filtering, particularly, matrix factorization. The seamless incorporation of geographical modeling into weighted regularized matrix factorization could be more beneficial. First, it better helps to cope with the sparsity challenges and location cold-start problems. Second, it helps to understand how to recommend locations when their geographical information provided, that is, how to automatically balance between geographical influence and personalized interest-based preference. To this end, we first propose GeoMF for joint geographical modeling and matrix factorization.

4.1 Optimization-based Kernel Density Estimation

Matrix factorization is a bilinear model, that is, given one mapping matrix fixed, the objective function is linear with respect to the other. In practice it resorts to optimization procedures like alternating least squares and stochastic gradient descent to learn mapping matrices. For the sake of its seamless incorporation with geographical modeling, we propose weighted linear regression for two-dimensional kernel density estimation in the task of geographical modeling. It involves two key concepts: user activity areas and location influential areas. Roughly speaking, user activity areas consist of spatial regions where the user will show up, and location influential areas are those spatial regions to which the influence of the location can be propagated. More specifically and formally, assuming the areas are obtained

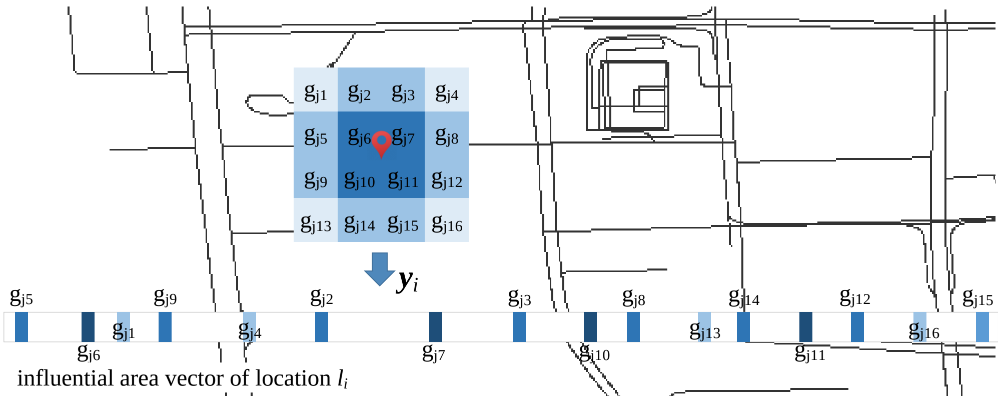


Fig. 1. Generating an influence area vector for a location l_i. The depth of color on each grid represents quantity of influence.

by splitting the whole world into F spatial grids of even-size, denoted as $\mathcal{G} = \{g_1, g_2, ..., g_F\}$, we have the following definitions:

Definition 4.1 (User Activity Areas). A user's activity areas include a set of spatial grids where the user may show up with non-negative possibilities.

We represent activity areas of a user a_u as a non-negative vector $\mathbf{x}_u = [x_{u,j}] \in \mathbb{R}^F_{\geq 0}$. Each entry $x_{u,j}$ indicates the possibility with which this user will appear in the grid $g_j \in \mathcal{G}$.

Definition 4.2 (Location Influential Areas). Influential areas of a location consist of a collection of spatial grids to which the influence of this location can be propagated.

We also represent influential areas of a location l_i by a non-negative vector $\mathbf{y}_i \in \mathbb{R}^F_{\geq 0}$, where each entry $y_{i,j}$ indicates how much influence is propagated to the spatial grid $g_j \in \mathcal{G}$ from the location l_i. Usually, influential areas are different from location to location. For simplicity, we assume the influence areas of a location are fixed in advance and have a normal distribution centered at this location. In particular, the influence propagated to a grid g_j from a location l_i is defined as $y_{i,j} = \frac{1}{\sigma} K(\frac{d_{i,j}}{\sigma})$, where $K(\cdot)$ is standard normal distribution, σ is standard deviation and $d_{i,j}$ represents geographical distance between location l_i and the center of the spatial grid g_j. Figure 1 shows an example of such a setting.

The advantage of setting the influential areas in this way is that inner product between $\mathbf{x}_u$ and $\mathbf{y}_i$ represents two-dimensional kernel density estimation on a user's visited locations. Especially, according to kernel density estimation, the density of a user a_u at a location l_i is estimated by

$$\hat{f}_\sigma(i) = \frac{1}{\sigma n_u} \sum_{l'_i \in \mathcal{L}} c_{u,i'} K(\frac{d_{i,i'}}{\sigma}). \tag{3}$$

where $n_u = \sum_{i'} c_{u,i'}$. If locations in $\mathcal{L}$ are mapped into spatial grids $\mathcal{G}$, this estimation becomes

$$\hat{f}_\sigma(i) = \sum_{g_j \in \mathcal{G}} \frac{n_{u,j}}{\sigma n_u} K(\frac{d_{i,j}}{\sigma}), \tag{4}$$

where $n_{u,j} = \sum_{l'_i \in \mathcal{L}} \delta(l_{i'} \in g_j) c_{u,i'}$ is the visit frequency of the user a_u to the spatial grid g_j while $\delta(l_{i'} \in g_j)$ means the location l_i falls into the spatial grid g_j. At this moment, by setting $\mathbf{x}_u$ being proportional to the visit frequency to the corresponding grid such that

$x_{u,j} = \frac{n_{u,j}}{\sigma n_u}$, the estimated density of the user a_u at the location l_i equals the inner product $\mathbf{x}_u^T \mathbf{y}_i$. For the sake of taking the characteristic of implicit feedback[1] into account, instead of assigning $\mathbf{x}_u$ empirical frequency, we treat it as a variable and learn it by optimizing the following objective function (dubbed GeoWLS [16]),

$$\min_{\mathbf{X}} \|\mathbf{W}^{\circ\frac{1}{2}} \odot (\mathbf{R} - \mathbf{X}\mathbf{Y}^T)\|_F^2 + \lambda\|\mathbf{X}\|_1, \quad \text{subject to } \mathbf{X} \geq 0 \tag{5}$$

where we stack activity area vector of each user by row to obtain a user activity area matrix $\mathbf{X} \in \mathbb{R}_{\geq 0}^{M\times F}$ and stack influential area vector of each location by row to obtain a location influential area matrix $\mathbf{Y} \in \mathbb{R}_{\geq 0}^{N\times F}$. $\|\mathbf{X}\|_1$ is a ℓ_1 norm of the matrix $\mathbf{X}$, encouraging a sparse solution for user activity areas [23]. The underlying reasons of imposing sparsity regularization on user activity areas are two-fold: first, users are usually constrained around several long-stay locations, such as, home or working places; second, it can also improve the effectiveness and the efficiency of recommendation, as shown in the previous conference paper [16].

4.2 Joint Model

Recall that in the matrix factorization model, the preference of the user a_u for the location l_i is expressed by $\mathbf{p}_u^T\mathbf{q}_i$, so geographical modeling becomes consistent with matrix factorization, making seamless combination possible. In particular, we leverage $\mathbf{X}$ and $\mathbf{Y}$ to respectively augment user latent factors $\mathbf{P}$ and location latent factors $\mathbf{Q}$ in the factorization model, as shown in Figure 2. Then the estimated preference matrix for the proposed GeoMF model is formulated as follows:

$$\hat{\mathbf{R}} = \left[\mathbf{P}, \mathbf{X}\right] \begin{bmatrix} \mathbf{Q} \\ \mathbf{Y} \end{bmatrix}^T = \mathbf{P}\mathbf{Q}^T + \mathbf{X}\mathbf{Y}^T \tag{6}$$

One reason for such an explicit augmentation with geographical information is that there is still no evidence showing that the latent space has already included them. In this way, a user's preference for a location is modeled as an inner product in the augmented space and thus includes both interest-based preference of the user from the latent space and geographical preference for the location. If geographical preference of one user for a location is non-zero, her activity areas intersect with the influential areas of the location so that the location is reachable from her activity areas.

4.3 Optimization

After augmentation, in addition to $\mathbf{P}$ and $\mathbf{Q}$, we are also required to learn $\mathbf{X}$ by minimizing the following objective function:

$$\min_{\mathbf{P},\mathbf{Q},\mathbf{X}} \|\mathbf{W}^{\circ\frac{1}{2}} \odot (\mathbf{R} - \mathbf{P}\mathbf{Q}^T - \mathbf{X}\mathbf{Y}^T)\|_F^2 + \gamma(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2) + \lambda\|\mathbf{X}\|_1, \quad \text{subject to } \mathbf{X} \geq 0 \tag{7}$$

The minimization of this objective function is achieved by an alternating optimization scheme. It consists of one procedure to take turn learning latent factors for users and locations when fixing $\mathbf{X}$, and another one involving sparse and non-negative weighted least squares with respect to $\mathbf{X}$ when fixing all latent factors. In each procedure, the objective function is non-increasing due to minimization, and thus the iteration of such an alternating

[1] The necessity has been evaluated in the previous conference paper [16]

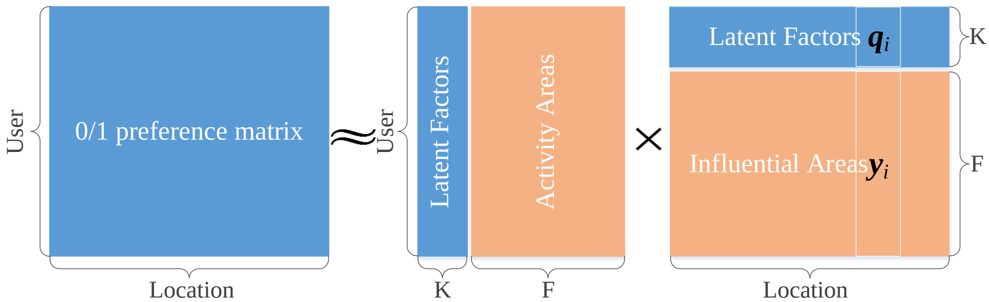


Fig. 2. The framework of GeoMF.

optimization can guarantee the non-increase of the objective function. Hence, such an alternating optimization algorithm can converge after some rounds of iterations.

4.3.1 Learning Latent Factors. When fixing user activity area matrix $\mathbf{X}$, the optimization of Eq (7) with respect to user/item latent factors is similar to alternating least squares in weighted regularized matrix factorization discussed previously. More specifically, the latent factor $\mathbf{p}_u$ of a user a_u, corresponding to the u-th row of $\mathbf{P}$, is updated based on

$$\mathbf{p}_u = (\mathbf{Q}^T\mathbf{W}^u\mathbf{Q} + \gamma\mathbf{I}_K)^{-1}\big(\mathbf{Q}^T\mathbf{W}^u(\mathbf{r}_u - \mathbf{Y}\mathbf{x}_u)\big) \tag{8}$$

where $\mathbf{W}^u$ is an $N \times N$ diagonal matrix, subject to $W^u_{i,i} = w_{u,i}$. Here, since we have set the same weight, i.e., 1 to the unvisited locations, there is a trick to speed up its calculation [11] by making use of $\mathbf{W}^u = \tilde{\mathbf{W}}^u + \mathbf{I}_N$ such that $\tilde{W}^u_{i,i}$ is non-zero only if $r_{u,i} \neq 0$. In particular, $\mathbf{Q}^T\mathbf{W}^u\mathbf{Q} = \mathbf{Q}^T\tilde{\mathbf{W}}^u\mathbf{Q} + \mathbf{Q}^T\mathbf{Q}$. In this case, the second part is independent of users so that it can be precomputed, costing $\mathcal{O}(NK^2)$, while the first part only requires $\mathcal{O}(\|\mathbf{r}_u\|_0K^2)$, being in proportion to the number of visited locations of the user a_u. Here ℓ_0 norm of matrix (vector) is the number of non-zero entries in this matrix (vector). For the inverse of a $K \times K$ matrix, we assume it requires $\mathcal{O}(K^3)$ time even though more efficient algorithms exist, particularly for a positive-semidefinite matrix, but probably are less relevant for the typically small values of K. Applying the similar trick to calculate the other part

$$\mathbf{Q}^T\mathbf{W}^u(\mathbf{r}_u - \mathbf{Y}\mathbf{x}_u) = \mathbf{Q}^T\mathbf{W}^u\mathbf{r}_u - \mathbf{Q}^T\tilde{\mathbf{W}}^u\mathbf{Y}\mathbf{x}_u - \mathbf{Q}^T\mathbf{Y}\mathbf{x}_u$$

where the right most term could be precomputed for all users, costing $\mathcal{O}\big((\|\mathbf{X}\|_0 + \|\mathbf{Y}\|_0)K\big)$. If first computing $(\tilde{\mathbf{W}}^u\mathbf{Y})\mathbf{x}_u$, the second term costs $\mathcal{O}(\|\mathbf{r}_u\|_0\|\mathbf{x}_u\|_0 + \|\mathbf{r}_u\|_0K)$. The first term is computationally less than another two terms, only costing $\mathcal{O}(\|\mathbf{r}_u\|_0K)$. Completing the final matrix multiplication between the inversed matrix and the resultant vector requires $\mathcal{O}(K^2)$ time. Therefore, it totally costs $\mathcal{O}\big(\|\mathbf{r}_u\|_0(K^2 + \|\mathbf{x}_u\|_0) + K^3\big)$ to update latent factors for the user a_u, without taking pre-computation overhead into account. When we update all users' latent factors in sequence, the overall worst-case time complexity is $\mathcal{O}\big(\|\mathbf{R}\|_0K^2 + MK^3 + (\|\mathbf{X}\|_0 + \|\mathbf{Y}\|_0)K + \|\mathbf{X}\|_0\|\mathbf{R}\|_{0,\infty}\big)$, where we borrow the notation of $\ell_{2,1}$ norm of matrices to denote $\|\mathbf{R}\|_{0,\infty} = \max_u \|\mathbf{r}_u\|_0$. It is worth noting that there is no independence of updating latent factors among different users, so that it is possible to resort to parallel updating for acceleration.

Similarly, we update the latent factor $\mathbf{q}_i$ of a location l_i, corresponding to the i-th row of $\mathbf{Q}$, by:

$$\mathbf{q}_i = (\mathbf{P}^T\mathbf{W}^i\mathbf{P} + \gamma\mathbf{I})^{-1}\big(\mathbf{P}^T\mathbf{W}^i(\mathbf{r}^i - \mathbf{X}\mathbf{y}_i)\big) \tag{9}$$

where $\mathbf{W}^i$ is an $M \times M$ diagonal matrix, subject to $W^i_{u,u} = w_{u,i}$. Applying the similar optimization trick, we can complete the update of latent factors for all locations in sequence in $\mathcal{O}\big(\|\mathbf{R}\|_0 K^2 + NK^3 + (\|\mathbf{X}\|_0 + \|\mathbf{Y}\|_0)K + \|\mathbf{Y}\|_0 \|\mathbf{R}^T\|_{0,\infty}\big)$.

In summary, the total complexity of updating latent factors in one iteration is $\mathcal{O}\big(\|\mathbf{R}\|_0 K^2 + (M+N)K^3 + \|\mathbf{X}\|_0(K + \|\mathbf{R}\|_{0,\infty}) + \|\mathbf{Y}\|_0(K + \|\mathbf{R}^T\|_{0,\infty})\big)$. It is easy to note that the sparsity structure of both $\mathbf{X}$ and $\mathbf{Y}$ is also important for the efficiency of updating these latent factors. Hence, at the same time of imposing ℓ_1 norm on the user activity area matrix $\mathbf{X}$, we also assume that two-dimensional normal distribution for generating location influential areas is truncated. In other words, only those areas within a certain threshold of distance (i.e., d km) from a location are considered its influential areas. This is reasonable to some extent since normal distribution usually decays quickly with the increase of the distance from its center.

4.3.2 Learning User Activity Areas. Now let's turn to learning the user activity area matrix $\mathbf{X}$. When fixing user/item latent factors, the objective function in Eq (7) with respect to $\mathbf{X}$ is similar to a sparse and non-negative weighted least squares problem, which can be further generalized as a bounded-variable least squares problem [12]. Such kinds of problems have been solved by several approaches, including active set method [12], sequential coordinate-wise algorithm [6], and projected gradient descent method [17]. Among these methods, projected gradient descent is highly efficient and has been extensively studied in non-negative matrix factorization, which can also be cast into two sub-problems related to non-negative least squares [17]. The general idea of the projected gradient descent algorithm is to update parameters by gradient descent and then to project the updated ones into feasible regions defined by bound constraints. Nevertheless, the choice of learning rate in gradient descent needs to guarantee that the projected parameters can sufficiently decrease the objective function in Eq (7). Thus we leverage the methods proposed in [17] to update the user activity area matrix. However, due to the existence of the weighting matrix, the gradient of Eq (7) with respect to $\mathbf{X}$ is a full matrix. It is impractical to update all the parameters at one time. Hence, we instead update each user's activity areas independently.

Let's rewrite the objective function with respect to activity area vector of a user a_u and discard the irrelevant terms,

$$\begin{aligned} \Omega(\mathbf{x}_u) = \|(\mathbf{W}^u)^{\circ\frac{1}{2}}(\mathbf{r}_u - \mathbf{Q}\mathbf{p}_u - \mathbf{Y}\mathbf{x}_u)\|_2^2 + \lambda\|\mathbf{x_u}\|_1 \\ \text{subject to } \mathbf{x}_u \geq 0 \end{aligned} \tag{10}$$

The gradient of $\Omega(\mathbf{x}_u)$ with respect to $\mathbf{x}_u$ is

$$\nabla\Omega(\mathbf{x}_u) = \underbrace{\mathbf{Y}^T\mathbf{W}^u\mathbf{Y}\mathbf{x}_u}_{\boldsymbol{\nabla}_1} + \underbrace{\lambda - \mathbf{Y}^T\mathbf{W}^u(\mathbf{r}_u - \mathbf{Q}\mathbf{p}_u)}_{\boldsymbol{\nabla}_2}. \tag{11}$$

Note that $\boldsymbol{\nabla}_1 = \mathbf{Y}^T\tilde{\mathbf{W}}^u\mathbf{Y}\mathbf{x}_u + \mathbf{Y}^T\mathbf{Y}\mathbf{x}_u$, where the first term costs $\mathcal{O}(\|\mathbf{r}_u\|_0\|\mathbf{x}_u\|_0 + \|\mathbf{r}_u\|_0 n_{\bar{i}})$ and the second term costs $\mathcal{O}(\|\mathbf{x}_u\|_0 F)$ by pre-computing $\mathbf{Y}^T\mathbf{Y}$. Here we assume there are the approximate same number of influential areas of different locations, denoted by $n_{\bar{i}}$. This could be true when the spatial grids are sufficiently small. And $\boldsymbol{\nabla}_2 = \lambda - \mathbf{Y}^T\mathbf{W}^u\mathbf{r}_u + \mathbf{Y}^T\tilde{\mathbf{W}}^u\mathbf{Q}\mathbf{p}_u + \mathbf{Y}^T\mathbf{Q}\mathbf{p}_u$, costing $\mathcal{O}(\|\mathbf{r}_u\|_0 n_{\bar{i}} + FK + \|\mathbf{r}_u\|_0 K)$ by pre-computing $\mathbf{Y}^T\mathbf{Q}$. Based on this gradient, we update $\mathbf{x}_u$ as follows:

$$\mathbf{x}_u^{(t+1)} = P_+(\mathbf{x}_u^{(t)} - \alpha\nabla\Omega(\mathbf{x}_u^{(t)})) \tag{12}$$

where the initial $\mathbf{x}_u^{(0)}$ is set as zero and $P_+(\mathbf{x})$ is a function to project a vector $\mathbf{x} \in \mathbb{R}^F$ onto its non-negative orthant $\mathbb{R}^F_{\geq 0}$. In particular,

$$P_+(x_j) = \begin{cases} x_j & \text{if } x_j > 0 \\ 0 & \text{otherwise} \end{cases}, \quad g_j \in \mathcal{G} \tag{13}$$

The learning rate, α is chosen so as to ensure the sufficient decrease of $\Omega(\mathbf{x}_u)$, i.e.,

$$\Omega(\mathbf{x}_u^{(t+1)}) - \Omega(\mathbf{x}_u^{(t)}) \leq \varepsilon \nabla\Omega(\mathbf{x}_u^{(t)})^T(\mathbf{x}_u^{(t+1)} - \mathbf{x}_u^{(t)}) \tag{14}$$

where ε is a parameter of this condition and commonly set as 0.01. Since $\Omega(\mathbf{x}_u)$ is a quadratic function with respect to $\mathbf{x}_u$, this condition can be quickly evaluated via the gradient and Hessian matrix ($\nabla^2\Omega(\mathbf{x}_u^{(t)}) = \mathbf{Y}^T\mathbf{W}^u\mathbf{Y}$), that is,

$$(1-\varepsilon)\nabla\Omega(\mathbf{x}_u^{(t)})^T\Delta\mathbf{x}_u^{(t)} + \frac{1}{2}\Delta\mathbf{x}_u^{(t)}{}^T\nabla^2\Omega(\mathbf{x}_u^{(t)})\Delta\mathbf{x}_u^{(t)} \leq 0 \tag{15}$$

where $\Delta\mathbf{x}_u^{(t)} = \mathbf{x}_u^{(t+1)} - \mathbf{x}_u^{(t)}$. In this case, in each step, although the objective function has decreased sufficiently, it requires repeatedly searching for a valid learning rate based on some heuristic rules. Assume $\#trials$ is the number of trials for searching the valid learning rate. Then during searching the valid learning rate, the following two observations could make it fast to compute projected gradient.

OBSERVATION 1. *If $x_{u,j}^{(t)} = 0$ and $[\nabla\Omega(\mathbf{x}_u^{(t)})]_j > 0$, then $x_{u,j}^{(t+1)} = 0, \forall t \in [0, \#trials]$ according to Eq (12), where $[\nabla\Omega(\mathbf{x}_u^{(t)})]_j \triangleq \frac{\partial\Omega(\mathbf{x}_u)}{\partial x_{u,j}}|_{\mathbf{x}_u=\mathbf{x}_u^{(t)}}$.*

OBSERVATION 2. *If $\mathbf{x}_u^{(0)} = \mathbf{0}$, then $x_{u,j}^{(t)} = 0, \forall t \in [2, \#trials]$ if $x_{u,j}^{(1)} = 0$.*

PROOF. If $\mathbf{x}_u^{(0)} = \mathbf{0}$, then $[\boldsymbol{\nabla}^2]_j > 0$ if $x_{u,j}^{(1)} = 0$, according to Observation 1. And following Eq (11), $[\nabla\Omega(\mathbf{x}_u^{(t)})]_j = [\boldsymbol{\nabla}^2]_j + \mathbf{y}_j^T\mathbf{W}^u\mathbf{Y}\mathbf{x}_u^{(t)} > 0, \forall t \in [1, \#trials]$ since each vector/matrix of $\mathbf{y}_j^T\mathbf{W}^u\mathbf{Y}\mathbf{x}_u^{(t)}$ is non-negative. Then recursively applying Observation 1, we can complete the proof. □

Based on Observation 2, it is obvious that the following observation could be satisfied.

OBSERVATION 3. *$\forall t \in [2, \#trials], \|\mathbf{x}_u^{(t)}\|_0 \leq \|\mathbf{x}_u^{(1)}\|_0$.*

However, there is no deterministic relationship between $\|\mathbf{x}_u^{(t)}\|_0$ and $\|\mathbf{x}_u^{(t+1)}\|_0$. This is because we may not only add relevant activity areas but also remove some redundant activity areas. By comparing kernel density estimation in Fig 8(a) of one sample user's visited locations with Fig 8(c) which plots user activity area learned from GeoMF, we can clearly see this two possibilities.

And note that $\|\mathbf{x}_u^{(1)}\|_0$ depends on the parameter λ, so λ determines the sparsity level of activity areas of each user. Hence it further determines the time complexity of computing activity areas for each user. In the following, for the sake of making complexity analysis easy, we assume $\|\mathbf{X}\|_0$ is known in advance.

Complexity Analysis. The time complexity analysis of gradient computation in Eq (11) shows that it costs $\mathcal{O}(\|\mathbf{r}_u\|_0\|\mathbf{x}_u\|_0 + \|\mathbf{x}_u\|_0 F + \|\mathbf{r}_u\|_0 n_{\bar{i}} + FK + \|\mathbf{r}_u\|_0 K)$ without pre-computation overhead. Due to $\|\Delta\mathbf{x}_u^{(t)}\|_0 \leq \|\mathbf{x}_u^{(1)}\|_0, \forall t \geq 2$, the evaluation of sufficient decrease condition doesn't cost as much as the computation of projected gradient. Assume that $\#iter$ is the number of iterations for projected gradient descent, when we perform an updating operation for each user in sequence (it can be done in parallel), the overall complexity is $\mathcal{O}\big(\#iter \times \#trials \times \big(\|\mathbf{X}\|_0\|\mathbf{R}\|_{0,\infty} + \|\mathbf{X}\|_0 F + MFK + \|\mathbf{R}\|_0(n_{\bar{i}} + K)\big)\big)$. Here, we once again observe the sparse structure of user activity areas plays an important role in improving efficiency of learning algorithms.

Connection with Kernel Density Estimation. When users' preference for locations are not taken into account and activity area vector $\mathbf{x}_u$ of a user a_u is initialized to zero, $\mathbf{x}_u^{(1)} = \alpha P_+(\mathbf{Y}^T\mathbf{W}^u\mathbf{r}_u - \lambda)$. Therefore, after the first iteration, activity areas of the user a_u includes the regions that can be directly reached from the user's visited locations by means of $\mathbf{Y}$. And the possibility of showing up in a spatial grid depends on the visit frequency via the weighting matrix $\mathbf{W}^u$. Thus, the update in this first iteration is similar to kernel density estimation except that it is subject to a sufficient decrease of $\Omega(\mathbf{x}_u)$. In the subsequent iterations, her activity areas are expanded or shrunken due to $\mathbf{Y}^T\mathbf{W}^u\mathbf{Y}$ under the condition of decreasing $\Omega(\mathbf{x}_u)$. It is worth noting that $\mathbf{Y}^T\mathbf{W}^u\mathbf{Y}$ actually encodes the personalized spatial correlation between grids.

5 GEOMF++

In spite of excellent explainability, GeoMF suffers from computational issues when the number of spatial grids (F) is large, according to the analysis of time complexity. To overcome the computational issues, we further propose GeoMF++ by mapping each spatial grid into a low-dimensional latent space. The subsequent analysis to GeoMF++ reveals its strong connection with two other widely-used algorithms for joint geographical modeling and matrix factorization.

5.1 Loss Function

For making GeoMF++ more flexible and scalable, each spatial grid is simply mapped into the same latent space as that formed by weighted regularized matrix factorization, as shown in Fig 3. Therefore, we can not only add them into location latent factor but also take their dot product with user latent factor to represent user preference for spatial grids. Denoting the mapping matrix from spatial grids to the latent space is $\mathbf{V} \in \mathbb{R}^{F\times K}$, user preference matrix $\mathbf{R}$ is then estimated by

$$\hat{\mathbf{R}} = \mathbf{P}\mathbf{Q}^T + \mathbf{P}(\mathbf{Y}\mathbf{V})^T = \mathbf{P}(\mathbf{Q} + \mathbf{Y}\mathbf{V})^T = [\mathbf{P}, \mathbf{P}]\begin{bmatrix}\mathbf{Q}\\ \mathbf{Y}\mathbf{V}\end{bmatrix}^T$$

Such an estimation could have the following three ways of explanation. First, each user's preference for locations does not only include both interest-based preference but also geographical preference. Second, each location latent factor includes random effect of both location itself and influential areas. Third, similar to GeoMF, we augment location latent factor with $\mathbf{YV}$, a linear mapping image of $\mathbf{Y}$. However, we don't augment user latent factor with $\mathbf{XV}$ since $\mathbf{X}$ is unknown. In principle, we can augment user latent factor with any matrix of the same size as $\mathbf{P}$. Choosing $\mathbf{P}$ can reduce the number of parameters and thus decrease model complexity of GeoMF++. Besides, it also simplifies the optimization procedure, as shown in the next subsection.

Based on such an estimation of the preference matrix, we then formulate the objective function as follows:

$$\min_{\mathbf{P},\mathbf{Q},\mathbf{V}} \|\mathbf{W}^{\circ\frac{1}{2}} \odot (\mathbf{R} - \mathbf{P}(\mathbf{Q} + \mathbf{V}\mathbf{Y})^T)\|_F^2 + \gamma_P\|\mathbf{P}\|_F^2 + \gamma_Q\|\mathbf{Q}\|_F^2 + \eta\|\mathbf{V}\|_F^2, \tag{16}$$

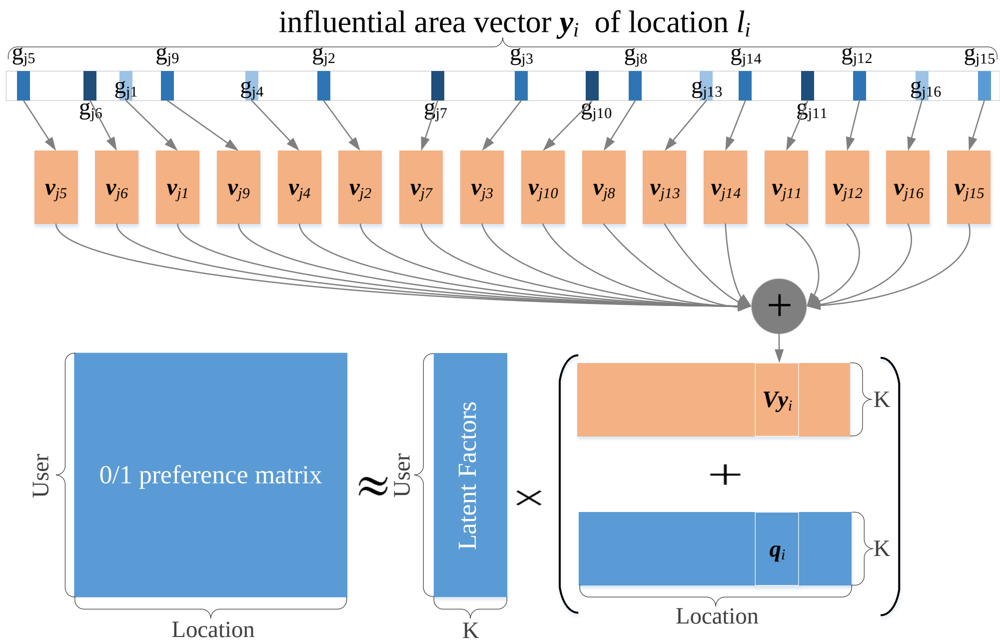


Fig. 3. The framework of GeoMF++.

Note that compared to WRMF, the regularized coefficient of $\mathbf{P}$ is distinguished from that of $\mathbf{Q}$, since $\mathbf{Q}$ should not be varied as much as $\mathbf{P}$ when there is auxiliary spatial information provided for locations. This is very close to feature-based matrix factorization [2] except that GeoMF++ has taken the characteristics of implicit feedback into account in a better way. Following the suggestion in [14], we introduce another matrix variable $\hat{\mathbf{Q}} = \mathbf{Q} + \mathbf{YV}$, and then simplify Eq (16) as

$$\min_{\mathbf{P},\hat{\mathbf{Q}},\mathbf{V}} \|\mathbf{W}^{\circ\frac{1}{2}} \odot (\mathbf{R} - \mathbf{P}\hat{\mathbf{Q}}^T)\|_F^2 + \gamma_P\|\mathbf{P}\|_F^2 + \gamma_Q\|\hat{\mathbf{Q}} - \mathbf{YV}\|_F^2 + \eta\|\mathbf{V}\|_F^2. \tag{17}$$

Now it is close to regression-based latent factor models [1], and in this case the optimization can be easily achieved by alternating least squares with respect to these three parameters. When convergent, we can employ $\mathbf{Q} = \hat{\mathbf{Q}} - \mathbf{YV}$ to obtain the random effect of locations themselves.

5.2 Optimization

Fixing $\hat{\mathbf{Q}}$, we derive the gradient of Eq (17) with respect to $\mathbf{p}_u$ and set it to zero. Due to being quadratic with respect to $\mathbf{p}_u$, we can obtain the closed update formulation for latent factor of the user a_u,

$$\mathbf{p}_u = (\hat{\mathbf{Q}}^T\mathbf{W}^u\hat{\mathbf{Q}} + \gamma_P\mathbf{I}_K)^{-1}\hat{\mathbf{Q}}^T\mathbf{W}^u\mathbf{r}_u \tag{18}$$

Similarly, fixing $\mathbf{P}$ and $\mathbf{V}$, we can derive the closed updating formulation for latent factor $\hat{\mathbf{q}}_i$ of the location l_i as follows:

$$\hat{\mathbf{q}}_i = (\mathbf{P}^T\mathbf{W}^i\mathbf{P} + \gamma_Q\mathbf{I}_K)^{-1}(\mathbf{P}^T\mathbf{W}^i\mathbf{r}^i + \gamma_Q\mathbf{V}^T\mathbf{y}_i) \tag{19}$$

Here we see that $\hat{\mathbf{q}}_i$ indeed captures the effect of its visit history and influential areas, whose balance is determined by γ_Q.

Finally, fixing $\hat{\mathbf{Q}}$, we can derive the gradient with respect to $\mathbf{V}$, and set it to zero. Then the updating formulation for $\mathbf{V}$ involves solving the system of linear equations as follows:

$$(\mathbf{Y}^T\mathbf{Y} + \frac{\eta}{\gamma_Q}\mathbf{I}_F)\mathbf{V} = \mathbf{Y}^T\hat{\mathbf{Q}}, \tag{20}$$

If directly inversing a sparse matrix $\mathbf{Y}^T\mathbf{Y}$ of size $F \times F$ for solving the system of linear equations, we may still suffer from computational issues. Hence, a much more practical solution is to leverage conjugate gradient descent, since it does not explicitly pre-compute and store the coefficient matrix $\mathbf{Y}^T\mathbf{Y}$, and only depends on the multiplication between these matrices.

Complexity Analysis. According to previous analysis, the time complexity of updating each row of $\mathbf{P}$ in sequence according to Eq (18) and each row of $\hat{\mathbf{Q}}$ in sequence according to Eq (19) is $\mathcal{O}\big(\|\mathbf{R}\|_0K^2 + (M+N)K^3 + \|\mathbf{Y}\|_0K\big)$. The time complexity of conjugate gradient descent for solving the system of linear equations is $\mathcal{O}(\|\mathbf{Y}\|_0K\#iter)$, where $\#iter$ is the number of iterations of conjugate gradient descent to reach a given threshold of approximation error. Therefore, the overall complexity of GeoMF++ at each iteration is $\mathcal{O}\big(\|\mathbf{R}\|_0K^2 + (M+N)K^3 + \|\mathbf{Y}\|_0K\#iter\big)$. It is worth noting that $\mathbf{p}_u$ and $\hat{\mathbf{q}}_i$ could be updated by entry-wise coordinate descent [9], so that the inverse of $K \times K$ matrix could be avoided. Hence, based on entry-wise coordinate descent algorithms, the time complexity of each iteration is $\mathcal{O}\big(\|\mathbf{R}\|_0K + (M+N)K^2 + \|\mathbf{Y}\|_0K\#iter\big)$, but it may require more iterations for convergence. Besides, more importantly, there is no dependence of updating latent factors among users and among locations given $\mathbf{V}$ fixed, so that parallel computing techniques could be exploited for further speedup.

5.3 Recovering User Activity Areas

Since $\hat{\mathbf{q}}_i$ indeed captures the effect of its visit history and influential areas, the dependence of $\mathbf{p}_u$ on $\tilde{\mathbf{Q}} = [\tilde{\mathbf{q}}_1, \cdots, \tilde{\mathbf{q}}_N]$ will lead to the dependence of $\mathbf{p}_u$ on not only her visited locations but also all influential areas of her visited locations. Hence, it is possible to recover user activity areas $\mathbf{x}_u$ from $\mathbf{p}_u$ via the following sparse and non-negative least squares problem:

$$\begin{aligned} &\min_{\mathbf{x}_u} \frac{1}{2}\|\mathbf{p}_u - \mathbf{V}^T\mathbf{x}_u\|_2^2 + \beta\sum_i x_{u,i} \\ &\text{subject to } x_{u,i} \geq 0, \forall l_i \in \mathcal{L} \end{aligned} \tag{21}$$

Here we can still leverage projected gradient descent for learning user activity areas $\mathbf{x}_u$. One example of user activity areas recovered from her latent factors has been shown in Fig 8(d). By comparing it with kernel density estimation in Fig 8(a) of this user's visited locations, we see that the recovered user activity areas are meaningful and reasonable to some extent. It is worth mentioning that $\mathbf{x}_u$ is not used for location recommendation any more, thus the efficiency of learning activity areas for all users doesn't affect training efficiency of GeoMF++.

5.4 Connection with Other Models

We have shown the relationship of GeoMF++ with feature-based matrix factorization [1, 2], so that location influential areas are actually considered as features of locations. In the following, we will show its strong connection with another two widely-used models for joint geographical modeling and matrix factorization. Both of them take geographical influence into account by incorporating distance between locations. However, distance has been required to convert similarity for further use. In particular, according to [13, 22], the similarity $s_{i,i'}$ between locations l_i and $l_{i'}$ is usually computed as follows:

$$s_{i,i'} = \begin{cases} e^{-\frac{d_{i,i'}^2}{4\sigma^2}}, & \text{if } d_{i,i'} < \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

where $d_{i,i'}$ denotes the distance between them, and $4\sigma^2$ is used instead of σ^2 for conveniently establishing the connection. Note that because similarity decays exponentially with the increase of squared distance, it sets a cut-off point ϵ to not take distant locations into account. Importantly, this makes similarity matrix $\mathbf{S} = [s_{i,i'}]$ sparse, and thus reduces the time and space complexity of any algorithm based on the similarity matrix.

5.4.1 Connection with Neighbor Additive Models. The first model estimates the preference matrix $\mathbf{R} = [r_{u,i}]$ by $\hat{r}_{u,i} = \tilde{r}_{u,i} + \sum_{i'} s_{i,i'}\tilde{r}_{u,i'}$, where $\tilde{r}_{u,i} = \mathbf{p}_u^T\mathbf{q}_i$. Hence, it is a hybrid model which combines content-based filtering with collaborative filtering. Rewriting the estimated preference in a matrix form, we have $\hat{\mathbf{R}} = \mathbf{P}(\mathbf{Q} + \mathbf{SQ})^T$, so that user preference for a location takes its neighbor locations into account in a linear way. Hence, we call it Neighbor Additive Models (NAM).

To establish the connection of GeoMF++ with NAM, we first derive a close relationship between the similarity matrix $\mathbf{S}$ and the location influential area matrix $\mathbf{Y}$, which could be stated in the following theorem.

THEOREM 5.1. *Assume the influential areas are restricted within $r = \frac{\epsilon}{2}$ from locations. If $e^{-\frac{\epsilon^2}{4\sigma^2}} \to 0$, then $\mathbf{y}_i^T\mathbf{y}_{i'} \to \frac{1}{2\Delta A}s_{i,i'}$, where ΔA is the area of any grid* [2].

PROOF. If the distance between two locations is larger than ϵ, their influential areas don't intersect with each other, so $\mathbf{y}_i^T\mathbf{y}_{i'} = 0 = \frac{1}{2\Delta A}s_{i,i'}$. Hence the approximation is exact. Otherwise, the dot product between them corresponds to the summation of $y_{i,j}y_{i',j}$ over any grid g_j within the intersection of their respective influential areas, as shown in Fig 4(a). Due to symmetric, we split it into four parts and focus on a upper right corner Ω.

$$\begin{aligned} \mathbf{y}_i^T\mathbf{y}_{i'} &= \frac{2}{\Delta A\pi\sigma^2}\sum_{g_j\in\Omega} e^{-\frac{d_{i,j}^2+d_{i',j}^2}{2\sigma^2}}\Delta A \\ &\approx \frac{2}{\Delta A\pi\sigma^2}\iint_\Omega e^{-\frac{d_{i,j}^2+d_{i',j}^2}{2\sigma^2}}dxdy \end{aligned}$$

where the last approximation is based on assumption that ΔA is sufficiently small. Then we transfer the Cartesian coordinate system to a polar coordinate system, which is established at the middle of two locations as in Fig 4(a). Assuming grid l_j located at angle θ, its maximum distance from origin is $\rho(\theta) = \frac{\sqrt{4r^2 - d_{i,i'}^2\sin^2\theta} - d_{i,i'}\cos\theta}{2}$, which is determined by

[2]The grid size should be sufficiently small so that the density within one grid is almost the same.

using $\rho(\theta)^2 + (\frac{d_{i,i'}}{2})^2 + d_{i,i'}\rho(\theta)\cos\theta = r^2$

$$
\begin{aligned}
\mathbf{y}_i^T\mathbf{y}_{i'} &= \frac{2}{\Delta A\pi\sigma^2}\int_0^{\frac{\pi}{2}} d\theta \int_0^{\rho(\theta)} \rho e^{-\frac{2\rho^2+\frac{d_{i,i'}^2}{2}}{2\sigma^2}} d\rho \\
&= \frac{1}{\Delta A\pi} e^{-\frac{d_{i,i'}^2}{4\sigma^2}} \int_0^{\frac{\pi}{2}} d\theta \int_0^{\rho(\theta)} e^{-\frac{\rho^2}{\sigma^2}} \frac{1}{\sigma^2} d\rho^2 \\
&= \frac{1}{\Delta A\pi} e^{-\frac{d_{i,i'}^2}{4\sigma^2}} \int_0^{\frac{\pi}{2}} d\theta \int_0^{\rho(\theta)} e^{-\frac{\rho^2}{\sigma^2}} \frac{1}{\sigma^2} d\rho^2 \\
&= \frac{1}{\Delta A\pi} e^{-\frac{d_{i,i'}^2}{4\sigma^2}} \int_0^{\frac{\pi}{2}} (1 - e^{-\frac{\rho(\theta)^2}{\sigma^2}}) d\theta \\
&= \frac{1}{\Delta A\pi} e^{-\frac{d_{i,i'}^2}{4\sigma^2}} \Big(\frac{\pi}{2} - e^{-\frac{r^2}{\sigma^2}} \int_0^{\frac{\pi}{2}} e^{-\frac{\alpha(\theta)}{4\sigma^2}} d\theta\Big)
\end{aligned}
$$

where $\alpha(\theta) = d_{i,i'}^2 \cos 2\theta - 2d_{i,i'}\cos\theta\sqrt{4r^2 - d_{i,i'}^2\sin^2\theta}$.

Since $e^{-\frac{r^2}{\sigma^2}} \to 0$, $\mathbf{y}_i^T\mathbf{y}_{i'} \to \frac{1}{2\Delta A} e^{-\frac{d_{i,i'}^2}{4\sigma^2}} = \frac{1}{2\Delta A} s_{i,i'}$ □

From the theorem, it directly follows that $\frac{\mathbf{y}_i^T\mathbf{y}_{i'}}{\|\mathbf{y}_i\|_2\|\mathbf{y}_{i'}\|_2} \to s_{i,i'}$. In order to see how well the similarity could be approximated, we compare them empirically using grids of size about $250m \times 250m$. The empirical results in Fig 4(b) show that they are consistent with the theoretical results.

If we rewrite this theorem via matrices, we have

$$\mathbf{S} \approx 2\Delta A\mathbf{Y}\mathbf{Y}^T. \tag{23}$$

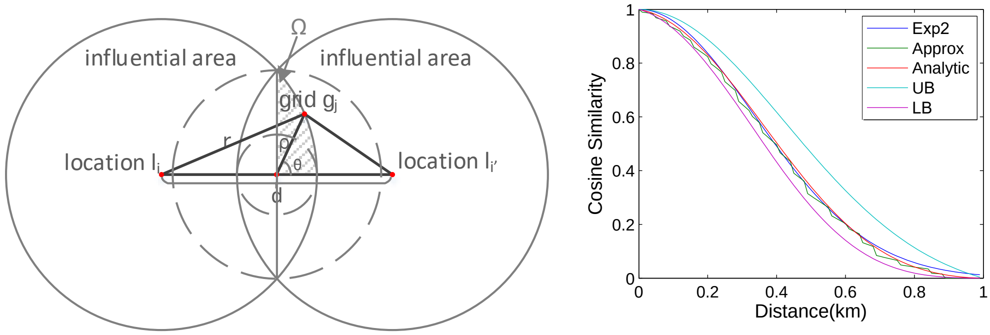
This means that we can approximate the similarity $\mathbf{S}$ via location influential areas $\mathbf{Y}$. Based on this decomposition of $\mathbf{S}$, we will show that GeoMF++ is strongly connected with NAM. In particular, if we introduce a new matrix $\mathbf{V} = 2\Delta A\mathbf{Y}^T\mathbf{Q}$ in the NAM, then $\hat{\mathbf{Q}} \overset{def}{=} \mathbf{Q} + \mathbf{S}\mathbf{Q} = \mathbf{Q} + \mathbf{Y}\mathbf{V}$, and the objective function of NAM becomes

$$
\begin{gathered}
\min_{\mathbf{P},\hat{\mathbf{Q}},\mathbf{V}} \|\mathbf{W}^{\circ\frac{1}{2}} \odot (\mathbf{R} - \mathbf{P}\hat{\mathbf{Q}}^T)\|_F^2 + \gamma_P\|\mathbf{P}\|_F^2 + \gamma_Q\|\hat{\mathbf{Q}} - \mathbf{Y}\mathbf{V}\|_F^2 \\
\text{subject to } (\mathbf{Y}^T\mathbf{Y} + \frac{1}{2\Delta A}\mathbf{I}_F)\mathbf{V} = \mathbf{Y}^T\hat{\mathbf{Q}}
\end{gathered}
$$

Note that the equality constraint is equivalent to $\min_{\mathbf{V}} \|\hat{\mathbf{Q}} - \mathbf{Y}\mathbf{V}\|_F^2 + \frac{1}{2\Delta A}\|\mathbf{V}\|_F^2$. Therefore, if alternating optimization is exploited after scalarizing this two-objective problem, that is to update $\mathbf{P}$, $\hat{\mathbf{Q}}$ and $\mathbf{V}$ in turns in NAM, it is almost the same as GeoMF++ except that the regularization coefficient of $\|\mathbf{V}\|_F^2$ changes.

5.4.2 Connection with Graph Regularized Models. In graph regularized models, the similarity matrix $\mathbf{S}$ is used for constructing graph Laplacian regularizer, $\mathrm{tr}(\mathbf{Q}'(\mathbf{D} - \mathbf{S})\mathbf{Q}) = \frac{1}{2}\sum_{i,i'} s_{i,i'}\|\mathbf{q}_i - \mathbf{q}_{i'}\|_2^2$, such that latent factors of more similar locations are closer to each other. Here $\mathbf{D}$ is a diagonal matrix, subject to $d_{i,i} = \sum_{i'} s_{i,i'}$. Therefore, extending WRMF, the graph Laplacian regularized model, called GWMF, optimizes the following objective functions:

$$\min_{\mathbf{P},\mathbf{Q}} \|\mathbf{W}^{\circ\frac{1}{2}} \odot (\mathbf{R} - \mathbf{P}\mathbf{Q}^T)\|_F^2 + \gamma(\|\mathbf{P}\|_F^2 + \gamma\|\mathbf{Q}\|^2) + \xi\mathrm{tr}(\mathbf{Q}^T(\mathbf{D} - \mathbf{S})\mathbf{Q}), \tag{24}$$



(a) (b)

Fig. 4. (a) The illustration for proving Theorem 5.1. (b) The comparison of cosine similarity of influential area vector between locations l_i and l_i' v.s. distance with $s_{i,i'}$ (Exp2). "Analytic" exploits numeric computation of integral while 'Approx' directly uses the dot product of influential area vector. "LB" and "UB" corresponds to its lower bound and upper bound by changing the integral zone to the inner dot circle and outer dot circle in (a).

Setting the gradient with respect to $\mathbf{q}_i$ to zero, we can derive the analytic solution for updating $\mathbf{q}_i$ when $\mathbf{p}_u$ fixed,

$$\mathbf{q}_i = (\mathbf{P}^T\mathbf{W}^i\mathbf{P} + (\gamma + \xi d_{i,i})\mathbf{I}_K)^{-1}(\mathbf{P}^T\mathbf{W}^i\mathbf{r}^i + \xi\mathbf{Q}^T\mathbf{s}_i), \tag{25}$$

Recall that in GeoMF++, the updating formulation of $\mathbf{V}$ has a closed form, so that it can be substituted back to the updating formulation of $\hat{\mathbf{q}}_i$. After applying matrix inversion lemma, we finally get

$$\hat{\mathbf{q}}_i = (\mathbf{P}^T\mathbf{W}^i\mathbf{P} + \gamma_Q\mathbf{I}_K)^{-1}(\mathbf{P}^T\mathbf{W}^i\mathbf{r}^i + \gamma_Q\mathbf{Q}^T\hat{\mathbf{s}}_i),$$

where $\hat{\mathbf{s}}_i = (\mathbf{Y}\mathbf{Y}^T + \frac{\eta}{\gamma_Q}\mathbf{I}_M)^{-1}\mathbf{Y}\mathbf{y}_i \approx (\mathbf{S} + \frac{2\Delta A\eta}{\gamma_Q}\mathbf{I}_M)^{-1}\mathbf{s}_i$, indicating $\hat{\mathbf{S}} \approx (\mathbf{S} + \frac{2\Delta A\eta}{\gamma_Q}\mathbf{I}_M)^{-1}\mathbf{S}$. The relationship between $\mathbf{S}$ and $\hat{\mathbf{S}}$ can be more clear, by using eigenvalue decomposition of the real symmetric similarity matrix $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{\Lambda}$ is a diagonal matrix whose entries are the eigenvalues of $\mathbf{S}$, and $\mathbf{U}$ is an orthogonal matrix whose columns are the corresponding eigenvectors. Then $\hat{\mathbf{S}} = \mathbf{U}(\mathbf{\Lambda} + \frac{2\Delta A\eta}{\gamma_Q}\mathbf{I}_M)^{-1}\mathbf{\Lambda}\mathbf{U}^T$, thus it just shrinks the eigenvalues to [0,1).

6 EXPERIMENTS

6.1 Datasets

We evaluate the proposed algorithms on two location-based social network datasets. One is the publicly available Gowalla dataset [5], which contains *6,423,854* check-ins at *1,280,969* locations from *107,092* users, where each user has *60* check-ins and checks in at *37* locations on average. The other one is a self-crawled Jiepang dataset, which contains *3,464,798* check-ins at *213,684* locations from *55,650* users. This subset was first collected by crawling all Beijing points-of-interest (POIs) and then aggregating all check-ins from users who have checked in at any of these Beijing locations. In this dataset, each user made 62 check-ins on average and these check-ins are dispersed at 16 locations on average. In both datasets, we select locations that are visited by at least 10 users and users who have visited at least

10 distinct locations. The statistics of these two datasets after preprocessing are shown in Table 1.

Table 1. Data statistics of two datasets after preprocessing

Dataset	#check-ins	#users	#locations	density
Gowalla	3,487,258	72,953	131,328	1.75e-04
Jiepang	2,985,136	29,763	41,222	1.30e-03

6.2 Evaluation Framework

In the evaluation, we investigate the effectiveness and efficiency of the proposed algorithms for incorporating geographical information. Following [30], we conduct evaluation from the perspectives of in-matrix recommendation and out-of-matrix recommendation. The former task, corresponding to a warm-start scenario, can be addressed by collaborative filtering techniques, while the latter task corresponds to the well-known location cold-start problem in recommendation and cannot resort to collaborative filtering.

In-matrix recommendation. In this evaluation, we use 5-fold cross-validation. For each user, we evenly split her all user-location pairs into 5 folds. We then iteratively consider each fold to be a test set and aggregate the other folds to be a training set. For each fold, we train the proposed models and baselines on the training set, and test the generated top-p recommendations against the within-fold locations for each user. We calculate recommendation performance of testing within each fold and report average performance metrics.

Out-of-matrix recommendation. This evaluation, corresponding to location cold-start problems, considers the case where a new collection of locations appear and no user has ever visited them. In this case, we evenly split all locations into 5 folds. For each fold of locations, we remove visiting history to them and train the recommendation algorithms on the remaining visiting history. After that, we test the top-p recommendations for each user against within-fold locations. We also calculate recommendation performance of testing within each fold and report average performance metrics.

6.3 Evaluation Measures

The learned model is then assessed by its capacity of finding the ground truth locations for each user among the top-p recommended locations. We exploit Recall and NDCG at a cut-off p for measuring such a capacity. Formally, if we denote the top-p recommended locations by $\mathbb{S}_u(p)$, the visited locations of user a_u by $\mathbb{V}_u$, and whether the recommendation at the position k is in the test set by $rel_{u,k}$

$$\begin{aligned} Recall@p &= \frac{1}{M}\sum_{u=1}^{M}\frac{|\mathbb{S}_u(p)\cap\mathbb{V}_u|}{|\mathbb{V}_u|} \\ NDCG@p &= \frac{1}{M}\sum_{u=1}^{M}\frac{DCG_u@p}{IDCG_u@p}, \end{aligned} \tag{26}$$

where $DCG_u@p = \sum_{k=1}^{p}\frac{2^{rel_{u,k}}-1}{\log_2(k+1)}$ and $IDCG_u@p$ is considered a normalization constant so that a perfert ordering (i.e. ranking locations by relevance scores) gets $NDCG_u@p$

score 1 for the user a_u. Compared to Recall, NDCG puts large emphasis on the top-ranked recommendation and better suits for assessing top-k recommendation performance. However, Recall, without discounting ranking positions, could be also used for assessing the recommendation performance of long-tail locations.

6.4 Parameters Setting

All these parameters are set by 5-fold cross validation. $w_{u,i} = 1 + \log(1 + c_{u,i} \times 10^{\alpha})$ according to [11]. After searching α over $\{1, 10, 30, 50, 100, 500, 1000, 5000, 10000\}$, $\alpha = 30$ on the Jiepang dataset and $\alpha = 1000$ on the Gowalla dataset. Note that when $\alpha >= 500$, this function could not be accurately evaluated due to floating overflow, we simply set $w_{u,i} = \alpha$ since it is almost not affected by visit frequency. The dimension of the latent space is set as 150 for all factorization-based algorithms without exception after searching K over $\{50, 100, 150, 200, 250, 300\}$. We also study the effect of latent space dimension K. We search γ_Q over $\{1000, 5000, 10000, 50000, 100000, 500000, 1000000\}$, and set them as $50,000$ and $500,000$ for the Jiepang dataset and for the Gowalla dataset, respectively. The γ in GeoMF and γ_P in GeoMF++ is insensitive, and simply set as 0.01. λ in GeoMF is set as 10 on the Jiepang dataset and 50 on the Gowalla dataset. The spatial grids are of size about $250m \times 250m$, and considered influential areas of a location if they are within $d = 1km$. Since λ, grid size and distance threshold have been studied in the conference paper, we do not present more results about them.

6.5 Baselines

We have proposed GeoMF and GeoMF++[3] for joint geographical modeling and matrix factorization, and compare them with the following baselines.

- **GWMF**, is the graph Laplacian regularized matrix factorization model, which has been introduced in Section 5.4.2. The efficient of graph Laplacian regularizer is searched over $\{0.1, 0.5, 1, 5, 10\}$.
- **WRMF** [11], weighted regularized matrix factorization, which doesn't take geographical modeling into account.
- **IRenMF** [22], a state-of-the-art location recommendation algorithm according to [21] and is considered Neighbor Additive Models in Section 5.4.1. For fair comparison, locations within 2km are considered similar in IRenMF since GeoMF++ takes influential areas within 1km as input. Other parameters are set as default values.
- ***UCF*** [33], has been studied in location recommendation, where the similarity between users is related to the number of their common visited locations and the preference of a user for a location is 0/1, indicating whether the user has visited the location. In other words, *UCF* is built based on $\mathbf{R}$.
- ***HPF*** [8], is a hierarchical Poisson-based matrix factorization and extended from non-negative matrix factorization by additionally places hierarchical gamma prior on user/item factors. We use their source code in Github and follow the default settings for parameters.
- **BPRMF** [27], also proposed for recommendation from implicit feedback datasets, optimizes a pairwise ranking objective function for learning latent factors. Its important parameters including learning rate and regularized coefficients are tuned by 5-fold cross validation.

[3]The source codes could be downloaded via https://github.com/DefuLian/recsys.git.

6.6 Results

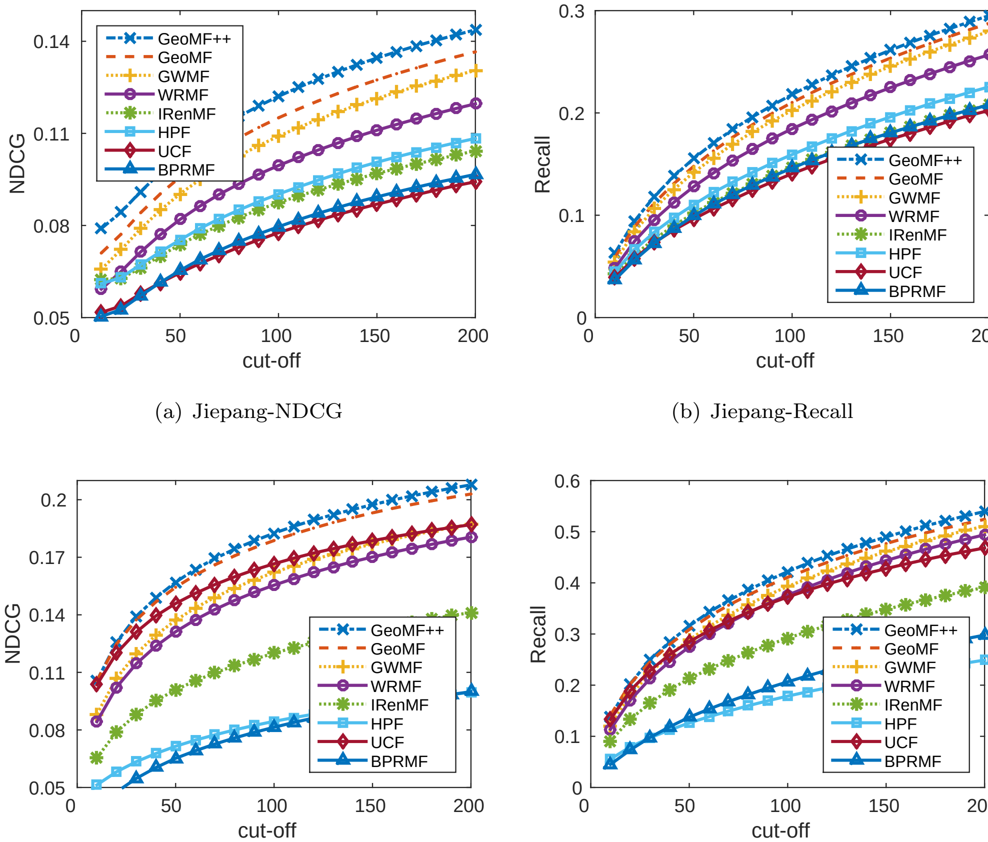
6.6.1 In-matrix Recommendation. The evaluation results of in-matrix recommendation are shown in Fig 5, where we set K as 50 for GeoMF on the Jiepang dataset, according to the trend of its performance with respect to the increase of K in Fig 6(a). From these four figures, we have the following observations.

First, GeoMF++ consistently outperforms GeoMF on both datasets in terms of both Recall and NDCG. This may not only lie in the dimension reduction techniques to figure out the semantic similarity of spatial grids, but also the usage of simple yet much more efficient optimization algorithms. More importantly, the application of dimension reduction techniques allows geographical modeling to be better aligned with matrix factorization, compared to GeoMF. However, interestingly, as shown in Fig 6(a) and Fig 6(c), when K is small, GeoMF is still better than GeoMF++. This may arise from a much larger dimension of augmented latent space in GeoMF. With the increase of K, since more useful information could be captured, GeoMF++ gradually becomes better and better than GeoMF. Such an observation aligns with that GeoMF approaches to and even surpass GeoMF++ when more user-location interaction data are used, as shown in Fig 6(b) and Fig 6(d). It is worth noting that on the Gowalla dataset, with the increase of K, it is empirically better to decrease the γ_Q. One of the reasons is data sparsity of the Gowalla dataset, so we need put more emphasis on memorizing more interaction history between users and locations when using a larger dimension of latent space. This point is consistent with gradual improvement of recommendation performance of WRMF with the increase of K on the Gowalla dataset. In contrast, due to higher density of the Jiepang dataset, there is little improvement of recommendation performance with the increase of K, so that GeoMF++ almost cannot benefit from lowering γ_Q down when increasing K.

Second, GWMF could not perform as well as GeoMF and GeoMF++. Due to the coupling of learning location latent factor, GWMF is sensitive to the order of updating. The commonly-used random order may make GWMF suboptimal. Besides, according to Fig 6(c), though the recommendation performance of both GWMF and WRMF improves with the increase of K, WRMF becomes better than GWMF when $K > 200$. This may indicate that fixing the coefficient of graph Laplacian regularizer along the whole training process may also lead to the suboptimal solutions. Besides, the same coefficient for graph Laplacian regularizer among all locations may be also suboptimal. In particular, when some locations have sufficient interaction data to determine their latent factors, its regularization coefficient should be smaller so that similar locations take less effect.

Third, IRenMF is also not as well as GeoMF and GeoMF++, and even worse than GWMF. One of important reasons is that the objective function is difficult to reduce when applying the accelerated proximal gradient (APG) for updating latent factors of all locations together, particularly when spatial similarity between locations is taken into account. This again validates the effectiveness of the proposed alternative optimization algorithm.

Forth, GeoMF++ and GeoMF outperform WRMF, indicating the benefit of knowledge of geographical information. The superiority of GeoMF++ to GeoMF indicates its more powerful capacity of geographical modeling within the matrix factorization framework. We have also tried to training geographical modeling and matrix factorization separately in GeoMF, in spite of not reporting the results, the recommendation performance is not as good as the joint approach in GeoMF. Hence, this illustrates the necessity of joint geographical modeling and matrix factorization.



(a) Jiepang-NDCG

(b) Jiepang-Recall

(c) Gowalla-NDCG

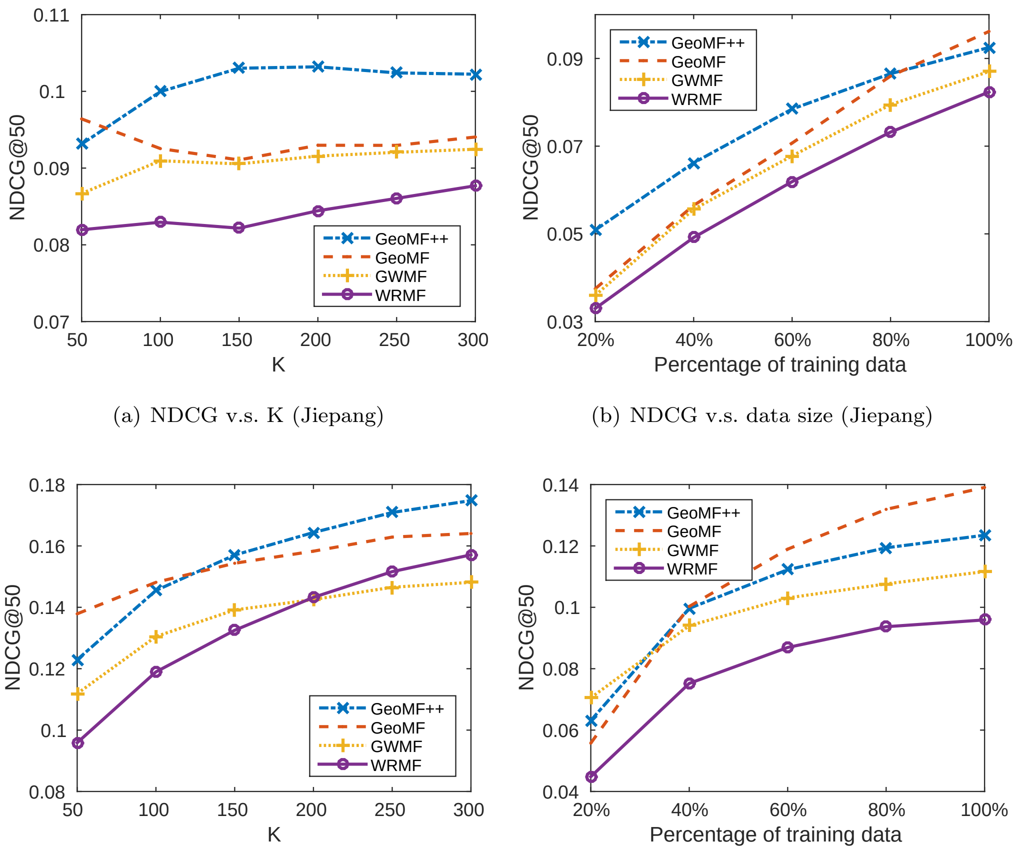
(d) Gowalla-Recall

Fig. 5. In-matrix evaluation results. Error bars are too small to show.

Fifth, user-based collaborative filtering (UCF) performs well compared to WRMF, in particular for the sparser Gowalla dataset. Comparing it with the results from the Jiepang dataset, it can be explained by that the sparser dataset requires a larger dimension of latent space for capturing more useful user-location interaction information. This is in line with the improvement of WRMF's recommendation performance on the Gowalla dataset with the increase of K.

Finally, WRMF outperforms HPF, and the gap between them is much larger on the Gowalla dataset. This is because HPF only models the observations while the Gowalla dataset is much sparser than the Jiepang dataset. In addition, WRMF is also better than BPRMF, though it exploits the ranking-based objective function and negative sampling techniques. In summary, WRMF works better for collaborative filtering from implicit feedback datasets than the other forms of matrix factorization. This provides the evidence that GeoMF and GeoMF++ are designed based on WRMF.

6.6.2 Out-matrix Recommendation. Although we have observed the superiority of GeoMF++ to GeoMF and other competing baselines in incorporating geographical modeling



(a) NDCG v.s. K (Jiepang)

(b) NDCG v.s. data size (Jiepang)

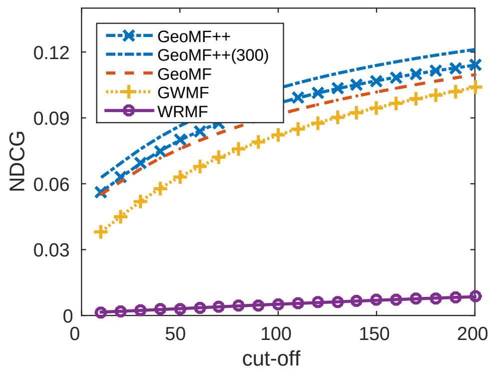
(c) NDCG v.s. K (Gowalla)

(d) NDCG v.s. data size (Gowalla)

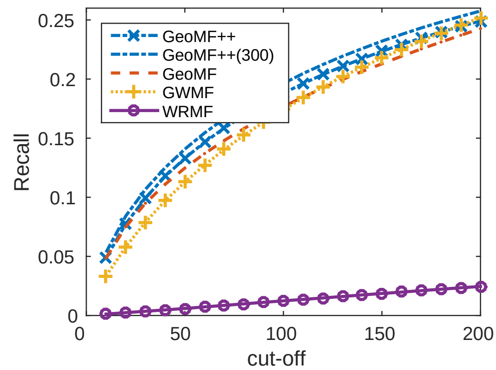
Fig. 6. Effect of latent space dimension and training data size.

into matrix factorization when we have user-location interaction histories, it is also interesting to investigate their performance difference in the out-matrix recommendation scenario. The results of comparison are shown in Fig 7, where IRenMF is not reported any more since it is even not as well as GWMF in case of in-matrix recommendation. In this case, WRMF and other collaborative filtering methods fail due to the lack of training data, and predicts users' preference score for items in a random way. Thus, the recommendation performance is close to zero in terms of Recall and NDCG. Note that, although there is no overlap of locations in the training set with that in the testing set, locations in the test set are still kept in the original user-location matrix. In this case, GeoMF and GeoMF++ could learn latent factors for testing locations based on influential areas, while GWMF could learn them based on similar locations. The results of comparison reveal the following interesting observations.

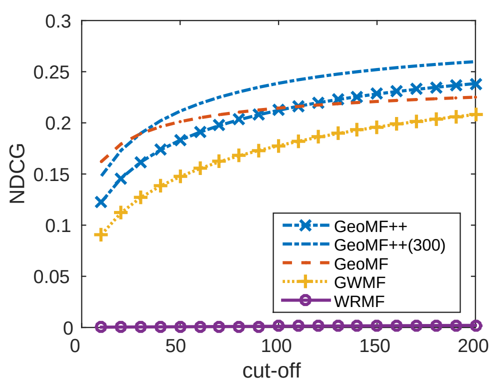
First, GeoMF++ can be better than GeoMF in the most cases when K=150, particularly on the denser Jiepang dataset. However, on the sparser Gowalla dataset, GeoMF is better than GeoMF++ in terms of NDCG with respect to the top-p recommendation when K=150. The major reason lies in its sparsity. The locations on the Gowalla dataset span over the whole America, so that its geographical density is much lower than the Jiepang dataset.



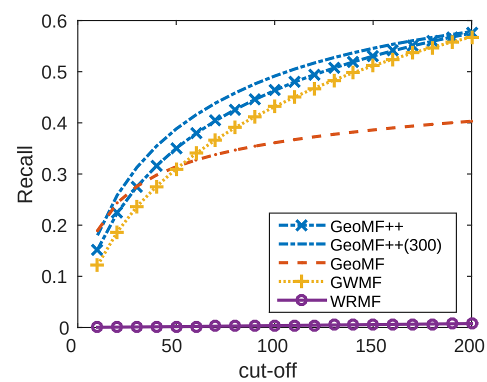
(a) Jiepang-NDCG



(b) Jiepang-Recall



(c) Gowalla-NDCG



(d) Gowalla-Recall

Fig. 7. Out-matrix evaluation results. Error bars are too small to show.

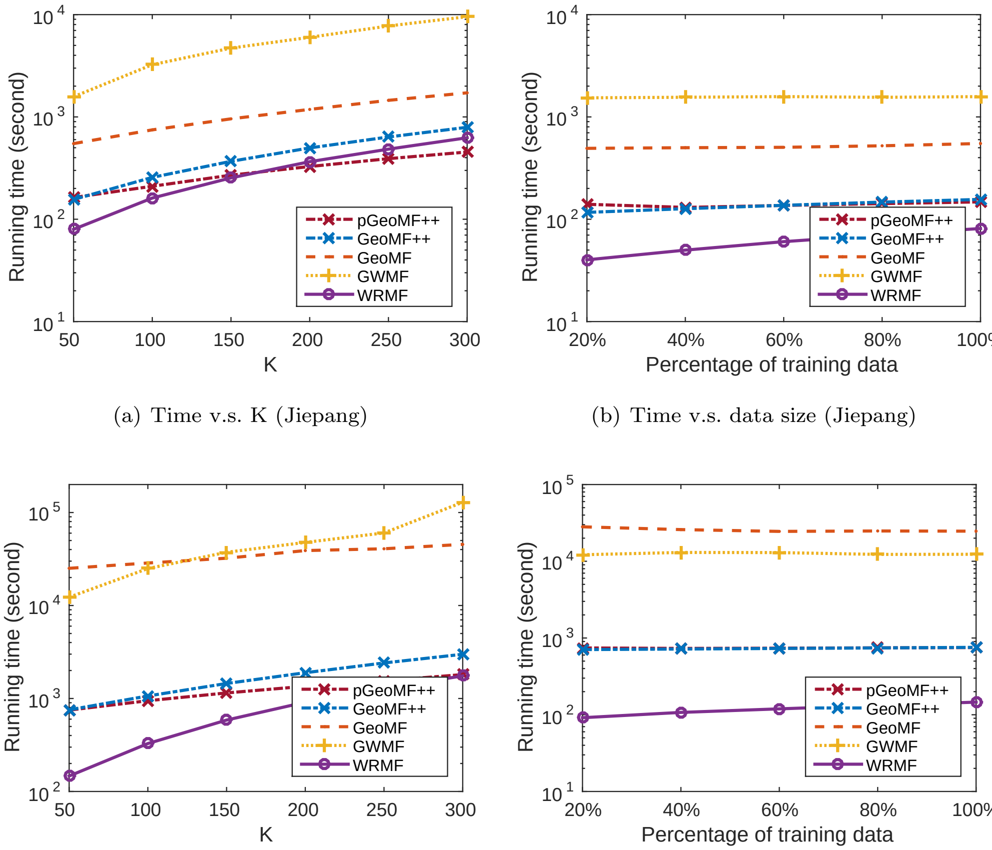
This leads to the necessity of a much larger dimension of latent space for capturing more geographical information using dimension reduction approaches, as shown by the comparison between GeoMF++(300) and GeoMF. Note that GeoMF is not affected by the dimension of latent space, since only the geographical modeling part in GeoMF takes effect.

Second, both GeoMF++ and GeoMF outperform GWMF in terms of both NDCG and Recall, again verifying the superiority of the proposed framework for joint geographical modeling and matrix factorization. Note that all regularization coefficients are simply set to the same values as in-matrix recommendation. This is because in practice it is not easy to collect sufficient new locations in location-based services to construct validation sets for fine tunning regularization coefficients, in particular after a long time of service running.

6.6.3 Interpretability. As pointed by [16], GeoMF is interpretable since its component-GeoWLS subsumes two-dimensional kernel density estimation. However, since GeoMF++ exploits embedded approaches for the sake of efficiency, it is unclear whether user latent factor indeed captures geographical information so that user activity areas can be accurately recovered. To this end, we randomly pick one Beijing user and plot kernel density estimation of

Fig. 8. Activity areas of one Beijing user using different algorithms based on the Jiepang dataset.

her visit history in Fig 8(a). The activity areas learned from GeoWLS, GeoMF and GeoMF++ are plotted in other three figures in Fig 8, respectively. Here, it is worth mentioning that for the sake of learning user activity robustly, the dimension K of latent space is set as 300. Compared to Fig 8(a), it seems that GeoWLS expands the areas of KDE while GeoMF shrinks the areas at the same time of expanding areas. Besides, GeoMF also puts more emphasis on some areas which the user has only visited a few times but may include many potentially attractive locations with respect to her. This also illustrates the joint approach of geographical modeling and matrix factorization can be beneficial. Although the difference of GeoMF++ from GeoMF is difficult to point out, user activity areas learned from GeoMF++ are almost around the KDE areas, and thus reasonable and meaningful to some extent. One certain observations is that the recovered activity areas from GeoMF++ tend to cluster together. Another interesting observation is that there emerge some areas which the user has never visited before. This mainly results from the dimension reduction techniques.



(a) Time v.s. K (Jiepang) (b) Time v.s. data size (Jiepang)

(c) Time v.s. K (Gowalla) (d) Time v.s. data size (Gowalla)

Fig. 9. Efficiency study.

6.6.4 The Study of Efficiency. We have shown that the recommendation performance has been significantly improved after resorting to dimension reduction techniques. In this section, we show the comparison of training efficiency. As shown in Fig 9, by comparing GeoMF++ with GeoMF on both two datasets, dimension reduction techniques greatly reduce the running time of training. Parallel computing achieves further speedup according to the comparison of pGeoMF++ with GeoMF++. Although GeoMF could also be accelerated by parallel computing, according to the superior efficiency of GeoMF to GWMF in Fig 9(a) and Fig 9(b), in the matlab environment it is easy to suffer from out-of-memory issues due to the necessity of replicating $\mathbf{Y}$ in multiple processes. That leads to the failure of parallel computing on the Gowalla dataset due to the large size of $\mathbf{Y}$, so that GeoMF is not as efficient as GWMF when K is small. In spite of this, when resorting to dimension reduction techniques, GeoMF++ becomes much more efficient than GWMF, because GeoMF++ uses a much smaller size of the matrix $\mathbf{Y}$ instead of location spatial similarity matrix $\mathbf{S} \approx 2\Delta A \mathbf{Y}\mathbf{Y}^T$. In spite of not reporting efficiency of IRenMF, it is easily presumed that it is at most the same as that of GWMF since IRenMF also uses location similarity matrix $\mathbf{S}$ instead of the matrix $\mathbf{Y}$. Moreover, GWMF can not leverage parallel computing techniques for acceleration due to the coupling of updating among different locations. Finally, according to these figures, the trend of running time with the increase of K and data size shows that GeoMF++ is pretty scalable, aligning with the analysis of time complexity. However, GeoMF++, GeoMF, and GWMF is almost invariant to the change of data size. This is because $\|\mathbf{R}\|_0$ is far smaller than $\|\mathbf{Y}\|_0$ on both datasets, so that the running time is dominated by updating parameters which depend on $\mathbf{Y}$.

6.7 Discussions

After resorting to dimension reduction techniques, GeoMF++ becomes a flexible framework, so that it could make use of other location features. Though the Gowalla dataset doesn't carry content information, we can leverage the time-stamps of visiting records to infer the category according to [32]. Hence, we feed visiting time distribution over 1-24 hours (distinguishing weekday from weekend) of each location as their features. However, the experimental results do not show significant improvement of recommendation performance in terms of Recall and NDCG. However, other metrics like AUC[27] and MPR (mean percentile rank)[11] can get improved on both datasets, indicating it can still take some effect. In the future, we can exploit more rich information from locations and even from user side for further improvement. However, when multiple types of content information are taken into account, we should carefully deal with their heterogeneity for better recommendation performance. Another interesting point is that, during searching regularization coefficients, we found the change of latent space dimension K may lead to the change of optimal regularization coefficients. Therefore, it is necessary to develop automatic hyper-parameter learning algorithms.

7 CONCLUSIONS

In this paper, based on previously developed GeoMF, we propose a scalable and flexible framework, dubbed GeoMF++, for joint geographical modeling and implicit feedback based matrix factorization. GeoMF++ improves the training efficiency of GeoMF by mapping location influential areas into the same low dimensional latent space as that formed by matrix factorization, and also improves the effectiveness of GeoMF since it leverages continuous latent factors to represent influential areas, so that it not only captures the semantic similarity between areas but also removes redundant and noisy information. By establishing the relationship of location influential areas with location spatial similarity matrix, GeoMF++ is strongly connected with neighbor additive models and graph Laplacian regularized models. GeoMF provides good interpretability since it subsumes two-dimensional kernel density estimation for geographical modeling, while GeoMF++ is still well explained by that user activity area can be recovered from user latent factors. The extensive evaluation on two large-scale LBSN datasets, reveals that 1) geographical modeling not only improves recommendation performance in the warm-start scenario, but also makes recommending cold-start locations possible; 2) both GeoMF++ and GeoMF outperform several competing baselines in terms of NDCG and Recall on both recommendation scenarios; 3) GeoMF++ is the most efficient algorithm compared to GeoMF and those baselines for joint geographical modeling and matrix factorization.

ACKNOWLEDGMENTS

The work is supported by the National Natural Science Foundation of China under Grant No.: 61502077 and 61631005.

REFERENCES

[1] Deepak Agarwal and Bee-Chung Chen. 2009. Regression-based latent factor models. In *Proceedings of KDD'09*. ACM, 19–28.

[2] Tianqi Chen, Weinan Zhang, Qiuxia Lu, Kailong Chen, Zhao Zheng, and Yong Yu. 2012. SVDFeature: a toolkit for feature-based collaborative filtering. *Journal of Machine Learning Research* 13, 1 (2012), 3619–3622.

[3] C. Cheng, H. Yang, I. King, and M.R. Lyu. 2012. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proceedings of AAAI'12*.

[4] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where you like to go next: successive point-of-interest recommendation. In *Proceedings of IJCAI'13*. AAAI Press, 2605–2611.

[5] E. Cho, S.A. Myers, and J. Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of KDD'11*. 1082–1090.

[6] Vojtěch Franc, Václav Hlaváč, and Mirko Navara. 2005. Sequential coordinate-wise algorithm for the non-negative least squares problem. In *Computer Analysis of Images and Patterns*. Springer, 407–414.

[7] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2013. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of RecSys'13*. ACM, 93–100.

[8] Prem Gopalan, Jake M Hofman, and David M Blei. 2015. Scalable Recommendation with Hierarchical Poisson Factorization. In *Proceedings of UAI'15. AUAI Press.*

[9] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of SIGIR'16*, Vol. 16.

[10] T. Horozov, N. Narasimhan, and V. Vasudevan. 2006. Using location for personalized POI recommendations in mobile environments. In *Proceedings of SAINT'06*. IEEE Computer Society.

[11] Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proceedings of ICDM'08*. IEEE, 263–272.

[12] Charles L Lawson and Richard J Hanson. 1995. *Solving Least Squares Problems*. Vol. 15. SIAM.

[13] Xutao Li, Gao Cong, Xiaoli Li, Tuan-Anh Nguyen Pham, and Shonali Krishnaswamy. 2015. Rank-GeoFM: A Ranking based Geographical Factorization Method for Point of Interest Recommendation. In *Proceedings of SIGIR'15*. ACM, 433–442.

[14] Defu Lian, Yong Ge, Fuzheng Zhang, Nicholas Jing Yuan, Xing Xie, Tao Zhou, and Yong Rui. 2015. Content-Aware Collaborative Filtering for Location Recommendation based on Human Mobility Data. In *Proceedings of ICDM'15*. IEEE, 261–270.

[15] Defu Lian, Xing Xie, Vincent W. Zheng, Nicholas Jing Yuan, Fuzheng Zhang, and Enhong Chen. 2015. CEPR: A Collaborative Exploration and Periodically Returning Model for Location Prediction. *ACM Trans. Intell. Syst. Technol.* 6, 1 (2015).

[16] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of KDD'14*. ACM, 831–840.

[17] Chih-Jen Lin. 2007. Projected gradient methods for nonnegative matrix factorization. *Neural computation* 19, 10 (2007), 2756–2779.

[18] Bin Liu, Yanjie Fu, Zijun Yao, and Hui Xiong. 2013. Learning geographical preferences for point-of-interest recommendation. In *Proceedings of KDD'13*. ACM, 1043–1051.

[19] Bin Liu and Hui Xiong. 2013. Point-of-Interest Recommendation in Location Based Social Networks with Topic and Location Awareness. In *Proceedings of SDM'13*. SIAM, 396–404.

[20] Xin Liu, Yong Liu, Karl Aberer, and Chunyan Miao. 2013. Personalized point-of-interest recommendation by mining users' preference transition. In *Proceedings of CIKM'13*. ACM, 733–738.

[21] Yiding Liu, Tuan-Anh Nguyen Pham, Gao Cong, and Quan Yuan. 2017. An experimental evaluation of point-of-interest recommendation in location-based social networks. *Proceedings of VLDB'17* 10, 10 (2017), 1010–1021.

[22] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. 2014. Exploiting Geographical Neighborhood Characteristics for Location Recommendation. In *Proceedings of CIKM'14*. ACM, 739–748.

[23] Kevin P Murphy. 2012. *Machine learning: a probabilistic perspective*. The MIT Press.

[24] Anastasios Noulas, Salvatore Scellato, Neal Lathia, and Cecilia Mascolo. 2012. A random walk around the city: New venue recommendation in location-based social networks. In *Proceedings of SocialCom'12*. IEEE, 144–153.

[25] R. Pan, Y. Zhou, B. Cao, N.N. Liu, R. Lukose, M. Scholz, and Q. Yang. 2008. One-class collaborative filtering. In *Proceedings of ICDM'08*. IEEE, 502–511.

[26] M.H. Park, J.H. Hong, and S.B. Cho. 2007. Location-based recommendation system using bayesian user's preference model in mobile devices. *Ubiquitous Intelligence and Computing* (2007), 1130–1139.
[27] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of UAI'09*. AUAI Press, 452–461.
[28] Y. Takeuchi and M. Sugimoto. 2005. An outdoor recommendation system based on user location history. In *Proceedings of the 1st International Workshop on Personalized Context Modeling and Management for UbiComp Applications*. Citeseer, 91–100.
[29] W.R. Tobler. 1970. A computer movie simulating urban growth in the Detroit region. *Economic geography* 46 (1970), 234–240.
[30] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of KDD'11*. ACM, 448–456.
[31] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhu Wang. 2013. A sentiment-enhanced personalized location recommendation system. In *Proceedings of HT'13*. ACM, 119–128.
[32] M. Ye, D. Shou, W.C. Lee, P. Yin, and K. Janowicz. 2011. On the semantic annotation of places in location-based social networks. In *Proceedings of KDD'11*. ACM, 520–528.
[33] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of SIGIR'11*. ACM, 325–334.
[34] Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. 2013. LCARS: a location-content-aware recommender system. In *Proceedings of KDD'13*. ACM, 221–229.
[35] Jia-Dong Zhang and Chi-Yin Chow. 2013. iGSLR: Personalized Geo-Social Location Recommendation-A Kernel Density Estimation Approach. In *Proceedings of GIS'13*.
[36] V.W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang. 2010. Collaborative filtering meets mobile recommendation: A user-centered approach. In *Proceedings of AAAI'10*. AAAl Press.
[37] V.W. Zheng, Y. Zheng, X. Xie, and Q. Yang. 2010. Collaborative location and activity recommendations with gps history data. In *Proceedings of WWW'10*. ACM, 1029–1038.
[38] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.Y. Ma. 2011. Recommending friends and locations based on individual location history. *ACM Trans. Web* 5, 1 (2011), 5.