

Location and Time Aware Social Collaborative Retrieval for New Successive Point-of-Interest Recommendation

Wei Zhang[†]

Jianyong Wang^{†‡}

[†]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

[‡]Jiangsu Collaborative Innovation Center for Language Ability, Jiangsu Normal University, Xuzhou 221009, China
zwei11@mails.tsinghua.edu.cn, jianyong@tsinghua.edu.cn

ABSTRACT

In location-based social networks (LBSNs), new successive point-of-interest (POI) recommendation is a newly formulated task which tries to regard the POI a user currently visits as his POI-related query and recommend new POIs the user has not visited before. While carefully designed methods are proposed to solve this problem, they ignore the essence of the task which involves retrieval and recommendation problem simultaneously and fail to employ the social relations or temporal information adequately to improve the results.

In order to solve this problem, we propose a new model called location and time aware social collaborative retrieval model (LTSCR), which has two distinct advantages: (1) it models the location, time, and social information simultaneously for the successive POI recommendation task; (2) it efficiently utilizes the merits of the collaborative retrieval model which leverages weighted approximately ranked pairwise (WARP) loss for achieving better top-n ranking results, just as the new successive POI recommendation task needs. We conducted some comprehensive experiments on publicly available datasets and demonstrate the power of the proposed method, with 46.6% growth in Precision@5 and 47.3% improvement in Recall@5 over the best previous method.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Information Filtering, Retrieval Models*

General Terms

Algorithms, Experimentation

Keywords

Collaborative Retrieval, Location based Social Networks, Successive Location Recommendation

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 ACM 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'15, October 19–23, 2015, Melbourne, VIC, Australia.
© 2015 ACM. ISBN 978-1-4503-3794-6/15/10 ...\$15.00.
DOI: <http://dx.doi.org/10.1145/2806416.2806564>.

1. INTRODUCTION

With the popularity of location-based social networks (LBSNs) allowing users to share their location-embedded information or attend location-aware activities at given time, more and more researchers pay attention to research tasks in this field, such as event-related recommendation [14, 28, 27], location prediction [12, 1], and location-aware information retrieval [22], etc.

In these studies, successive POI recommendation [4] is a newly emerging task which tries to treat users' current POIs as POI-related queries and recommend new POIs users have not visited before. Hence, it can be regarded as a hybrid task involving both retrieval and recommendation. It is important for real-time location-aware recommender systems. For example, it is extremely common that when a user visited one location just now, he may not be clear where to go next. The recommender system can treat the current location of the user as his location-aware query and automatically recommend new POIs, which is beneficial for both users and businesses.

Compared with traditional POI recommendation task which also tries to recommend new POIs for users [3, 26, 13, 7, 10, 9], the essential difference is that the performance of successive POI recommendation task is largely influenced by users' current visiting locations. To be more specific, the location which a user will visit next is usually near his current visiting POI. Besides, the shift from one specified location to another specific location may be very common. One example is the popular travel route in a tourist city, in which the transition from one POI to another is usually fixed. Without explicitly modeling the users' current visiting POIs, the approaches of traditional POI recommendation cannot capture the localization of successive visiting and the frequent shift patterns. Thus, it will result in poor performance by directly applying the models designed for traditional POI recommendation to successive POI recommendation task.

Existing approaches for successive POI recommendation task [18, 4] consider users' current locations to generate recommendation results. Nevertheless, they ignore to utilize the social relations between users or temporal information of visiting adequately, which can influence users' final decisions to a certain extent. On the one hand, social relations indicate that users may visit locations their friends have been to, which is reasonable as friends have similar interests. On the other hand, users and locations both have their own temporal characteristics. Different users may have various visiting patterns in the same period which are determined by their own habits. POIs also have different probability of being

visited in different time periods. For example, restaurants are more likely to be visited in lunch and dinner time. In later data analysis, we verify that check-in patterns of users and POIs over timelines are indeed diverse through correlation analysis. Besides, the characteristic of hybrid retrieval and recommendation involved in this task still needs to be studied, especially when modeling the top-k ranking results of recommendations which is significant for successive POI recommendation, as for each user the number of successive POIs visited for specific location-aware query is small.

To tackle the above issues, we propose a new model called *location and time aware social collaborative retrieval model (LTSCR)*. It is a natural extension of the recently proposed collaborative retrieval model (CRM) [21] which models query, user, and item in a unified model and first employs the loss function called weighted approximate rank pairwise (WARP) loss [19, 20] in collaborative filtering to gain better top k recommendation results and is more suitable for hybrid task involving retrieval and recommendation task simultaneously, which is similar to our successive POI recommendation task. In order to better model the users' successive check-in behaviors in LBSNs, LTSCR considers users' current visiting locations, visiting time, and user himself as implicit queries. Each candidate location is composed of three types of latent factors: (1) the first kind of factor captures transition patterns between successive check-in POIs, (2) the second models preference between users and locations, (3) and the third reveals locations' popularity over time. Besides, social relations are utilized to regularize the degree of interest similarity between friends, and the localization property is employed to constrain candidate POIs to be within the near regions of the query POI.

In summary, our major contributions in this paper are summarized as follows:

- We consider users' current visiting locations, visiting time, social relations, and the localization of successive check-ins simultaneously for successive POI recommendation task.
- We propose a complete model called *Location and Time Aware Social Collaborative Retrieval (LTSCR)* which incorporates the above information into a unified model and utilizes the merits of standard collaborative retrieval model to the successive POI recommendation task in LBSNs.
- We have conducted experiments on publicly available data sets from LBSNs. The results demonstrate that LTSCR performs better than several other methods.

The rest of the paper is organized as follows: we first discuss the related works in Section 2. Section 3 gives necessary preliminaries for successive modeling. In Section 4, we formulate our model in detail. Section 5 presents the experimental results and analysis. Finally, we conclude this paper in Section 6.

2. RELATED WORK

In this section, we briefly introduce three lines of researches related to our task: (a) traditional POI recommendation; (b) successive POI recommendation; and (c) collaborative retrieval.

Traditional POI Recommendation. Traditional POI recommendation task has been extensively studied in the last several years. Many researches hold a central assumption that nearer locations will be more likely to be visited by users. Based on this, a series of works incorporate distance influence factor into diverse traditional recommendation models [24, 3, 11]. Ye et al. [24] first combined distance influence factor with user-based collaborative filtering method. Then Cheng et al. [3] blended probabilistic matrix factorization with a gaussian mixture model to capture the influence of distance. More recently, Kurashima [11] proposed a geo topic model to integrate distance factor with latent Dirichlet allocation [2]. Another research direction is to utilize temporal information check-ins to improve location recommendation results [26, 7, 10]. Similar to the evolutionary trend of researches for integrating distance factor, they also incorporate temporal information into user-based collaborative filtering, matrix factorization and topic model successively. The above works demonstrate the advantages of considering location distance information and temporal information in traditional POI recommendation, which can motivate the research in successive POI recommendation.

Successive POI Recommendation. Lately, a few successive POI recommendation works [18, 4] have been conducted. Snag et al. [18] proposed a probabilistic model to integrate category transition probability and POI popularity to solve this problem. However, their method relies on categories of POIs and user attributes such as gender and residence information, which might not be easy to get due to privacy concern. Thus their method cannot be directly used in this paper. Cheng et al. [4] extended the state-of-the-art context-aware recommendation method, i.e., factorized personalized Markov chain (FPMC) [16], to provide a new model called FPMC with localized region constraint for successive POI recommendation. They divided the geographical space into grids, and the candidate recommendation locations for a user should be within the grid the user currently stays in or its surrounding eight grids. This condition is called localized region constraint. The experimental results indicated the effectiveness of their method. As they ignore to utilize the social relations between users and temporal information of visiting, our works first propose two extensions based on their model to incorporate these useful information and then we propose a new model called LTSCR to combine the extensions with the standard collaborative model, which makes our works largely different from theirs. Two recent works [5, 23] try to predict the next user movement based on users' previous sequential movements. Yet our problem is obviously different from theirs in two aspects. First, they consider which place users will move to and when they will reach while we concentrate on recommending new POIs which users have not visited before. Second, they need to utilize multiple recent check-ins of each user to construct sequential model for location transfer and time information prediction while we only regard each user's current visiting location as a query for collaborative retrieval. Besides, Yin et al. [25] proposed a location-content-aware recommender system which recommends items given a query location from users. However, their problem setting is different from ours as the query locations in theirs are regions (e.g. cities) and they want to recommend spatial items within those regions while our queries include all locations and we try to recommend POIs near those query locations.

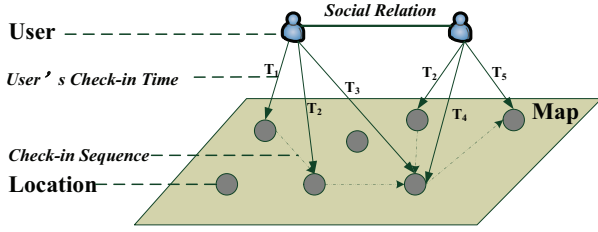


Figure 1: Basic Overview of Location-based Social Network.

Collaborative Retrieval. Collaborative retrieval task was first proposed by Weston et al. [21]. It blends traditional information retrieval and personalized recommendation tasks. In order to solve this problem, they proposed a latent collaborative retrieval model, which could be roughly divided into two parts. The first part models the relation among query, user, and item for solving retrieval problem. The second part captured the relation between user and item for dealing with the recommendation problem. Besides, the model adopts the WARP loss function to train the model in order to gain better top k results. Upon this model, Hsiao et al. [9] incorporated social relation information to regularize user latent factors and achieved better results. More details about collaborative retrieval model can be referred in the next section. In this paper, we propose a new model called LTSCR based on the standard collaborative model to solve our problem.

3. PRELIMINARIES

In this section, we first analyze properties of the datasets from LBSNs. Then we formulate the successive POI recommendation problem in our setting. After that, the standard method for this task is presented. Finally, the collaborative retrieval model is introduced at last as the cornerstone of the subject of the next section.

3.1 Data Analysis

Location-based social networks are common in current social networks. Almost all LBSNs and their variations share the following basic data attributes: user set \mathcal{U} , POI set \mathcal{P} , social relation set \mathcal{S} , and temporal information set \mathcal{T} . Their primary relations are shown in Figure 1. Basically, for a user u , when he visits a POI p at time t , he can share this check-in record with others, such as his friend set $S(u)$. In this work, we adopt two publicly available datasets published by [6] to test our proposed models. The first one was crawled from Brightkite and the other was from Gowalla. Thus, we denominate the first data set as Brightkite and the second as Gowalla for convenience.

In order to compute the distance distribution of successive check-in pairs, we first mine the successive check-in pairs. Inspired by the methods mentioned in [21, 4], we determine a maximal time interval T_{max} . For any two successive check-in POIs from the same user, the check-in time difference can not exceed T_{max} . After we collect enough consecutive check-in pairs, we calculate the cumulative distribution of distance between successive check-in locations. Figure 2 shows that in Gowalla dataset, 90.1% check-in pairs lie within 5 miles and only less than 6% pairs' distances surpass 10 miles. This phenomenon about localization of successive check-ins is less

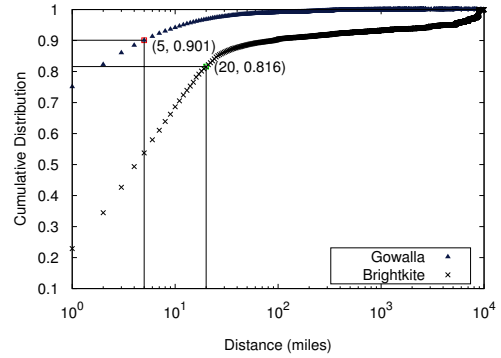


Figure 2: Distance Between Successive Check-in POIs.

visible in Brightkite dataset. However, there are still more than half check-in pairs' distances within 5 miles and more than eighty percents pairs' distances less than 20 miles. In summary, users' next POI choices are constrained in local regions near their current visiting POIs.

Besides, we also want to check whether temporal information may influence the successive POI recommendation results. It is intuitive that for different users, their own habits may cause very different visiting patterns as time goes on. And for specific POI, its check-in patterns should be different from other POIs belonging to different categories. For example, parks are usually visited by users in daytime, while restaurants are preferred by users at dinner time. To verify this, we utilize Pearson correlation to measure the correlation of users' temporal check-in patterns in the same location region to ensure the local time consistency and the same as for POIs. More specifically, we first filter users or locations with few check-in records. We use a 24 dimensional vector to represent the temporal check-in pattern of each user or item, with each dimension corresponding to an hour and each value denoting the total check-in count belonging to that hour. Then for users (or POIs) in the same geographical region, we calculate their Pearson correlation in pair and average the results finally for all pairs in all regions. The results for both datasets are shown in Figure 3. We can see that all of them lies in the interval $[-0.22, +0.32]$, which means the temporal check-in patterns have low positive correlation. Another phenomenon is that the correlation of locations' temporal check-in patterns is a little lower than users' on both datasets. In all, the check-in patterns of users and POIs over time are indeed diverse. The above analysis suggests us to utilize the temporal information of visiting to help successive POI recommendation.

3.2 Problem Formulation

Based on the above data analysis, we formulate the successive POI recommendation task below. Given a user $u (u \in \mathcal{U})$, his already visited POI set $P(u)$, his current visiting POI $i (i \in \mathcal{P})$, and the corresponding visiting timestamp $t (t \in \mathcal{T})$, the task is to recommend a new POI $c (c \notin P(u))$ for user to visit successively. Generally, top k new POIs are delivered to user u which are ranked according the score gotten from a defined score function $f(u, t, i, c)$.

In the collaborative retrieval setting, i and t can be considered as a location and time aware query of user u . The user wants to know which successive POI he can visit in the near

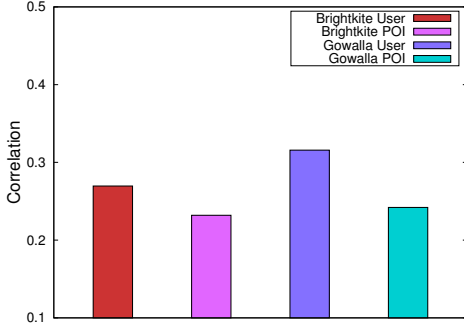


Figure 3: Average Correlation of Temporal Check-in Patterns of Users/POIs.

future. Hence, the above task can be regarded as one kind of collaborative retrieval task. However, there is no need to consider all POIs as candidate results as the works [21, 9] do because of the localization property of successive check-ins. Thus, we only need to search near POIs according to i to determine the final results, which is supported by the previous data analysis.

3.3 Collaborative Retrieval Modeling

The central idea of collaborative retrieval is to capture the relation among users, queries, and items in a unified model to calculate the possibility score $f(u, q, i)$ of user u preferring item i under the query q . Assume \mathcal{I} and \mathcal{Q} denote the item set and query set, respectively. \mathcal{K} is the dimension of latent factors, $\mathbf{U} \in \mathbb{R}^{\mathcal{K} \times |\mathcal{U}|}$ are user factors, $\mathbf{V} \in \mathbb{R}^{\mathcal{K} \times |\mathcal{I}|}$ are item factors, $\mathbf{Q} \in \mathbb{R}^{\mathcal{K} \times |\mathcal{Q}|}$ are query factors, $\mathbf{U}_u \in \mathbb{R}^{\mathcal{K}}$ is the latent factor of user u , $\mathbf{V}_i \in \mathbb{R}^{\mathcal{K}}$ is the latent factor of item i , and $\mathbf{Q}_q \in \mathbb{R}^{\mathcal{K}}$ is the latent factor of query q . Then for a user u , given a query $q (q \in \mathcal{Q})$ and an item i , the score function is formally expressed as follows,

$$f(u, q, i) = \mathbf{Q}_q^T \mathbf{T}_u \mathbf{V}_i + \mathbf{U}_u^T \mathbf{V}_i \quad (1)$$

where $\mathbf{T}_u \in \mathbb{R}^{\mathcal{K} \times \mathcal{K}}$ is a user dependent transform matrix. We can see that the first part of the equation is to capture the relation among query, user, and item, while the second part is to model the relation between user and item. Thus, the first part corresponds to the retrieval model and the second part corresponds to the recommendation model.

The collaborative retrieval model utilizes weighted approximate rank pairwise (WARP) [19] loss to complement traditional pairwise loss functions. Unlike previous pairwise loss functions which treat all training pairs equally, the WARP loss actively learns more from training pairs where positive items have high ranking positions and thus can achieve better top k ranking results (see [20] for more explanations). The rank for each positive item i is defined as below,

$$r(i) = \sum_{j \neq i} \mathbb{I}((1 + f(u, q, j)) > f(u, q, i)) \quad (2)$$

where $\mathbb{I}(\cdot)$ is an indicator function. It is clear that if $f(u, q, i)$ gets larger value, less items will satisfy $((1 + f(u, q, j)) > f(u, q, i))$ and item i should get lower rank.

Computing the scores of all items to get ranks is not efficient. Hence, an approximate method is proposed. For a given positive item i , it draws a negative item j with-

out replacement and terminates until item j satisfies the margin-based comparison $1 + f(u, q, j) > f(u, q, i)$. Suppose the number of sampling step to get j for i is $N_{s,i}$. Then the approximate rank is calculated below,

$$\tilde{r}(i) \approx \left\lfloor \frac{|\mathcal{I}| - 1}{N_{s,i}} \right\rfloor \quad (3)$$

where $\lfloor \cdot \rfloor$ means the floor function, $|\mathcal{I}|$ is the total number of items.

After getting the approximate rank, the rank loss for the positive item i is defined as follows,

$$L(\tilde{r}(i)) = \sum_{n=1}^{\tilde{r}(i)} \omega_n, \quad \text{with } \omega_1 \geq \omega_2 \geq \dots \geq 0. \quad (4)$$

where ω_n is the weight of position n in the ranked list and often defined to be $1/n$.

To optimize the loss function, efficient learning algorithms such as stochastic gradient descent (SGD) require that it is derivable for model parameters. Following the idea from [20], the margin function $\max(0, 1 + f(u, q, j) - f(u, q, i))$ can be added to meet the requirement. During the learning process, all the training pairs are weighed in every iteration according to the above two equations.

In summary, the collaborative retrieval model has two main characteristics: (1) it computes both query dependent and independent preference score through the retrieval model and recommendation model respectively, and jointly learn query, user, and item factors in a unified model; (2) it utilizes WARP loss function for optimization to get better top-k ranking recommendation results. Inspired by the above two ideas, we design our model for better modeling users' successive check-in behaviors and further achieve superior recommendation results.

4. METHODS

In this section, we first discuss the method to handle social relations and temporal information. Based on this, we then introduce the integrated new model we proposed and the corresponding model parameter learning approach.

4.1 Social Relation

Social relations are basic elements in social networks, which cover friendship, following relationship, trust relationship, etc. These relations could be one-way or two-way relation. Usually, users with social relations tend to have similar preferences and affect each other in some activities. Intuitively, these are also suitable for successive POI recommendation task, since users have more possibility of visiting the POIs their friends have visited.

Social regularization is an elegant approach to integrating social relations with the latent factor model [15]. It encourages users who are socially connected to have similar latent factors. Assume the original objective function is \mathcal{L}_o , the objective function of social regularization corresponds to \mathcal{L}_s , then the hybrid objective function becomes their blending, i.e., $\mathcal{L}_o + \alpha \mathcal{L}_s$ with a relative weight α .

We define the objective function of social regularization as follows,

$$\mathcal{L}_s = \sum_{u \in \mathcal{U}} \left\| \mathbf{U}_u - \frac{1}{\sum_{u' \in S(u)} \mathbf{B}_{u,u'}} \sum_{u' \in S(u)} \mathbf{B}_{u,u'} \mathbf{U}_{u'} \right\|_2 \quad (5)$$

where $S(u)$ means the friend set of user u , $\mathbf{B}_{u,u'}$ is the similarity value of user u and his friend u' , $\|\cdot\|_2$ is the L2-norm. Thus, the central problem becomes how to compute the similarity value $\beta_{u,u'}$. Using binary social relation directly cannot capture the different degrees of similarity between user u and his different friends. To address this issue, we define the following formula to compute the similarity degree,

$$\mathbf{B}_{u,u'} = \begin{cases} \frac{\sum_{i=1}^{|U|} S_{u,i} S_{u',i}}{\sqrt{\sum_{i=1}^{|U|} (S_{u,i})^2} \sqrt{\sum_{i=1}^{|U|} (S_{u',i})^2}} & \text{if } u' \in S(u) \\ 0 & \text{else} \end{cases} \quad (6)$$

where $S_{u,i}$ equals 1 if user u and i are friends. We only keep their friends who rank within top k by the similarity degree to save computation cost for later parameter learning in Equation 9. In the experiments, we set k to be 10.

4.2 Temporal Information

To model temporal information of users' check-ins, we first segment the continuous temporal space into fixed-sized time periods. For example, we can just treat one hour of a day as one time period, which means the number of time periods $|\mathcal{T}|$ is 24. Then for any timestamp, it will belong to one time period uniquely. This could be easily determined by the hour value of the timestamp. In addition to constructing time periods based on hours, we can also consider weekday, month, and season information. There is a trade-off between time granularity and model complexity. In other words, too many periods will result in the overfitting problem, and further influence the final results. In this paper, we find that dividing 24 hours of each day into 6 time periods and meanwhile differentiating weekdays and weekends can get a little better results. Thus the number of time periods is set to be 12 ($|\mathcal{T}| = 2 \times 6$) in experiments.

Based on previous data analysis, incorporating the time period information into the new model for both users and POIs is promising. To reflect the diverse preference of users in different time periods, we extend the user latent factor $\mathbf{U}_u \in \mathbb{R}^{\mathcal{K}}$ to the user temporal latent matrix $\mathbf{U}_u \in \mathbb{R}^{\mathcal{K} \times |\mathcal{T}|}$, where $\mathbf{U}_{u,t} \in \mathbb{R}^{\mathcal{K}}$ denotes the temporal latent factor in time period t for user u . Besides, in order to reveal the popularity of each location changed over time periods, we add a temporal popularity factor $\mathbf{T}_i \in \mathbb{R}^{|\mathcal{T}|}$ for each POI i .

One concern for the operations is that the user temporal matrix \mathbf{U}_u has $\mathcal{K} \times |\mathcal{T}|$ parameters to be learned. Thus it is more likely to suffer from overfitting than without considering temporal information. To overcome this issue, we can further add temporal regularization term to the objective function to control user temporal matrix in addition to social regularization term. In this work, we suppose that the change of factors of near time periods may be smooth. We calculate the similarity degree between near time periods based on the check-in behaviors of each user through cosine similarity. For each time period t , there are two near time periods, i.e., $t-1$ and $t+1$. The first time period is near to the last due to periodicity. We denote the two similarity values as $\mathbf{A}_{t,t-1}$ and $\mathbf{A}_{t,t+1}$, respectively. Assume $P(u,t)$ denotes the POI set user u has visited in time period t , then the similarity degree $\mathbf{A}_{t,t-1}$ can be computed as follows,

$$\mathbf{A}_{t,t-1} = \frac{\sum_{i=1}^{|P|} P_{u,t,i} P_{u,t-1,i}}{\sqrt{\sum_{i=1}^{|P|} (P_{u,t,i})^2} \sqrt{\sum_{i=1}^{|P|} (P_{u,t-1,i})^2}} \quad (7)$$

Analogously, $\mathbf{A}_{t,t+1}$ can be calculated in the same way.

4.3 Model Construction and Learning

Inspired by the above results and the collaborative retrieval model introduced in Section 3.3, we formulate our new model (LTSCR). For a user u , $\mathbf{U}_{u,t} \in \mathbb{R}^{\mathcal{K}}$ denotes his temporal latent factor. For a POI i , it has three types of latent factors: 1) $\mathbf{P}_i \in \mathbb{R}^{\mathcal{K}}$ captures the transition patterns between successive check-in POIs. It can be regarded as the retrieval part in the collaborative retrieval model. Adding additional user dependent transformation matrix like CRM for this task does not improve the final results significantly and increase the computation cost through our experiments. Thus we leave them out in our approach; 2) $\mathbf{V}_i \in \mathbb{R}^{\mathcal{K}}$ models the preference between users and locations. It can be considered as the recommendation part in CRM; 3) $\mathbf{T}_i \in \mathbb{R}^{|\mathcal{T}|}$ denotes the popularity of the location changed over time periods.

Given the above statement, the recommendation ranking score can be calculated as follows. For a user u , assume the current POI he is visiting is i and the corresponding time period is t , which means the input query of user u is constituted by a POI-time pair (i,t) . The preference to the next candidate POI c he may visit is measured by the following score function,

$$f(u,t,i,c) = (1 - \alpha - \beta) \mathbf{P}_i^T \mathbf{P}_c + \alpha \mathbf{U}_{u,t}^T \mathbf{V}_c + \beta \mathbf{T}_{c,t} \quad (8)$$

where α and β control the relative strength of each part above and should meet the constraints: $\alpha \in \mathbb{R}_+$, $\beta \in \mathbb{R}_+$, and $\alpha + \beta \leq 1$. The setting of these two parameters will be discussed in Section 5.4.2. We should emphasize that we adopt $\mathbf{P}_i^T \mathbf{P}_c$ instead of $\mathbf{P}_i^T \mathbf{V}_c$ in the above equation as it can achieve better experimental results. This may be explained by the fact that the first one can directly learn the transition patterns from data.

In order to utilize the localized region constraint, we constrain the candidate recommendation location c should be within the grid the user currently stays in (the grid i belongs to) or its surrounding eight grids, which is the same as [4]. More details about the grid size setting can be referred in the later dataset preprocessing section.

We incorporate the social regularization term and the temporal regularization term mentioned before into our new model for learning from users' real check-in behaviors by expanding the standard WARP loss function. Given a user u and his current visiting POI i , we denote his true successive visiting POI as c . The learning process should first sample one negative POI c' to construct the training pair (u,t,i,c') . Under the localized region constraint, all the negative POIs should also be within the grid covering i or its surrounding eight grids. We use $\Omega_{tr}(u)$ to denote all the training pairs related to user u . With this spirit, LTSCR solves the following learning problem,

$$\begin{aligned} & \min_{\mathbf{U}, \mathbf{P}, \mathbf{V}, \mathbf{T}} \sum_{u \in \mathcal{U}} \sum_{t \in \Omega_{tr}(u)} L(\tilde{r}(c)) \cdot \max(0, 1 + f(u,t,i,c') - f(u,t,i,c)) \\ & + \frac{\sigma}{2} \sum_{u \in \mathcal{U}} \sum_{t \in |\mathcal{T}|} \|\mathbf{U}_{u,t} - \frac{1}{\sum_{u' \in S(u)} \mathbf{B}_{u,u'}} \sum_{u' \in S(u)} \mathbf{B}_{u,u'} \mathbf{U}_{u',t}\|_2 \\ & + \frac{\gamma}{2} \sum_{u \in \mathcal{U}} \sum_{t \in |\mathcal{T}|} \mathbf{A}_{t,t-1} \|\mathbf{U}_{u,t} - \mathbf{U}_{u,t-1}\|_2 + \mathbf{A}_{t,t+1} \|\mathbf{U}_{u,t} - \mathbf{U}_{u,t+1}\|_2 \\ & + \frac{\lambda}{2} \sum_{i \in \mathcal{P}} (\|\mathbf{V}_i\|_2 + \|\mathbf{P}_i\|_2 + \|\mathbf{T}_i\|_2) \end{aligned} \quad (9)$$

where the sampled negative POI c' for query POI i with positive POI c in LTSCR should satisfy the margin-based comparison $1 + f(u, t, i, c') > f(u, t, i, c)$. The last terms of the above optimization target are regularization terms which control the overfitting problem and are often employed in latent factor models. The number of sampling step for finding the adequate negative POI determines the approximate rank $\tilde{r}(c)$ according to Equation 3. σ , γ , and λ are regularization parameters for the model which can be tuned on validation datasets.

The SGD algorithm is leveraged to optimize the above objective function. The central idea of SGD is to randomly scan all training pairs and update parameters through the gradient descent direction of the target function for each training pair. Each update is done by the following equation,

$$\Theta \leftarrow \Theta - \eta \cdot \frac{\partial Obj(\Theta)}{\partial \Theta} \quad (10)$$

where η corresponds to the learning rate. Θ denotes all model parameters, and Obj denotes the objective function shown in Equation 9.

Given a training pair (u, t, i, c, c') from the whole training data, the key gradients of all related parameters are computed as follows,

$$\begin{aligned} \frac{\partial Obj(\mathbf{U}_{u,t,k})}{\partial \mathbf{U}_{u,t,k}} &= \alpha(\mathbf{V}_{c',k} - \mathbf{V}_{c,k})L(\tilde{r}(c)) \\ &+ \sigma(\mathbf{U}_{u,t,k} - \frac{1}{\sum_{u' \in S(u)} \mathbf{B}_{u,u'}} \sum_{u' \in S(u)} \mathbf{B}_{u,u'} \mathbf{U}_{u',t,k}) \\ &+ \gamma(\mathbf{A}_{t,t-1}(\mathbf{U}_{u,t,k} - \mathbf{U}_{u,t-1,k}) + \mathbf{A}_{t,t+1}(\mathbf{U}_{u,t,k} - \mathbf{U}_{u,t+1,k})) \\ \frac{\partial Obj(\mathbf{P}_{i,k})}{\partial \mathbf{P}_{i,k}} &= (1 - \alpha - \beta)(\mathbf{P}_{c',k} - \mathbf{P}_{c,k})L(\tilde{r}(c)) + \lambda \mathbf{P}_{i,k} \\ \frac{\partial Obj(\mathbf{P}_{c,k})}{\partial \mathbf{P}_{c,k}} &= (1 - \alpha - \beta)(-\mathbf{P}_{i,k})L(\tilde{r}(c)) + \lambda \mathbf{P}_{c,k} \\ \frac{\partial Obj(\mathbf{V}_{c,k})}{\partial \mathbf{V}_{c,k}} &= -\alpha \mathbf{U}_{u,t,k}L(\tilde{r}(c)) + \lambda \mathbf{V}_{c,k} \\ \frac{\partial Obj(\mathbf{T}_{c,t})}{\partial \mathbf{T}_{c,t}} &= -\beta L(\tilde{r}(c)) + \lambda \mathbf{T}_{c,t} \end{aligned} \quad (11)$$

We only list the gradients related to the positive location c . However, it is easy to derive the similar gradients related to the negative location c' . After the gradients are calculated, we can update the parameters through Equation 10 uniformly. In summary, the complete learning algorithm for LTSCR is described in Algorithm 1.

5. EXPERIMENTAL STUDY

To demonstrate the effectiveness and efficiency of our proposed models, we conducted experiments on two real datasets. We first describe the data preprocessing procedure in Section 5.1 and then give a simple introduction to the evaluation metrics we used in this task. We explain all the methods we compare in the experiments in Section 5.3. Lastly, we compare the results of all methods and analyze the possible reasons. All the algorithms were implemented using C++ and we conducted all the experiments on a server (four 3.00GHz CPU cores, 20GB memory) with 64-bit Linux system. In this paper, we exploit the same ordinary data structures for all the algorithms and do not consider any parallel computation.

Algorithm 1: The Learning Algorithm

Input: Training records \mathcal{TR} , Social similarities \mathcal{SS} , Temporal similarities \mathcal{TS} , Geographical grids \mathcal{GG}
Output: Optimal user factors \mathbf{U} , POI factors $\mathbf{W} = \{\mathbf{P}, \mathbf{V}, \mathbf{T}\}$.

- 1 $tt=0$;
- 2 Initialize hyper-parameters;
- 3 Randomly initialize $\mathbf{U}, \mathbf{P}, \mathbf{V}, \mathbf{T}$;
- 4 **while** $Obj(\mathbf{U}, \mathbf{P}, \mathbf{V}, \mathbf{T})$ not converged and $tt \leq T_{max}$ **do**
- 5 Randomly shuffle the training records \mathcal{TR} ;
- 6 **foreach** training record k **do**
- 7 Get (u, t, i, c) from record k ;
- 8 Sampling desirable negative POI c' ;
- 9 Compute all gradients through Equation 11;
- 10 Update model parameters by Equation 10;
- 10 $tt = tt + 1$;

5.1 Datasets Preprocessing

We employ two public datasets, i.e., Brightkite and Gowalla in the experiments, just as mentioned in Section 3.1. As stated in Section 3.2, we address the problem of successive POI recommendation problem. When a user specifies a POI and its corresponding visiting time, the algorithms should recommend new POIs the user prefers to visit in the near future. In order to evaluate the results of all methods, we should extract reasonable successive check-in records from the same user to simulate the process. In the experiments, we regard two successive check-ins within $T = 2$ (hour) as a valid successive check-in pair. This strategy is similar to [21, 4]. To filter noisy data, we further select users who have at least 10 records and POIs which have at least 5 records in all check-in pairs gotten from the previous step. After finishing the preprocessing, we gain the ultimate experimental datasets. The details of the two datasets are shown in Table 1.

Table 1: Basic Statistics of Location-based Social Network Data.

Dataset	User	POI	Check-in Pair	Edge
Brightkite	4930	14393	126160	64504
Gowalla	34113	146544	1138971	283570

In order to utilize the localized region constraint motivated by the data analysis in Section 3.1, we divide the geographical space into nearly equally-sized grids whose areas are $0.05(\text{longitude}) \times 0.05(\text{latitude})$, which can include most successive check-in pairs. We randomly divide the two datasets into training, validation, and test datasets according to the ratio 7 : 1 : 2, meanwhile, assure that each user and POI occur at least one time in the training data. All the models are trained on the training datasets and all the hyper-parameters are tuned on the validation datasets. Finally, we compare all the models on the test datasets.

5.2 Evaluation Measures

To evaluate the successive POI recommendation results, we adopt two types of standard evaluation metrics: (1) Pre-

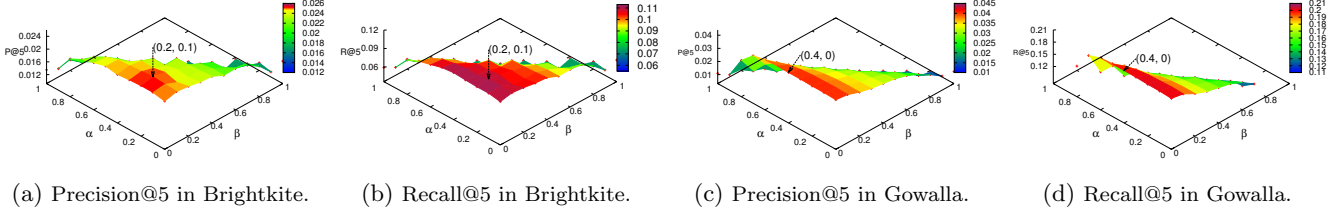


Figure 4: Impact of Different Settings of α and β .

recision@n (Precision at Position n) and Recall@n (Recall at Position n), and (2) MRR(Mean Reciprocal Rank).

Precision and recall are mainly used in previous recommendation tasks [21, 4] which relate to our work. Generally, it is computed by the following formula,

$$R@n = \frac{N_{tr}}{N_t}, \quad P@n = \frac{N_{tr}}{n} \quad (12)$$

where N_{tr} denotes the number of truly recommended items, and N_t represents the number of total true items.

We should emphasize that for each query POI, the number of true recommendation results is small for collaborative retrieval task. Therefore, the precision of results may be relative low as long as k is not small. That's also why the works [21, 9] do not use metric Precision@n. As a complementary, we adopt MRR metric that concentrates on measuring positions of the first correct recommendation results. In our problem setting, MRR is calculated as

$$MRR = \frac{1}{|U|} \sum_u \frac{1}{|Q_u|} \sum_i \frac{1}{rank_i} \quad (13)$$

where $|Q_u|$ is the number of implicit location queries each user u has and $rank_i$ specifies the position of the first correct successive POI the user will visit based on the query i .

5.3 Comparison Methods and Components

We adopt the following five methods in the experiments. Among them, the first two are popularity based methods and the middle two are model based methods. The main targets of the experiments are to demonstrate that our proposed method outperforms these methods.

- **POP**: Pure popularity based method is simple with only considering the total check-in count of each POI for this task. Thus it can be regarded as the baseline for our later comparisons.
- **POP+LR**: It briefly extends POP by additional incorporating localized region constraint for POI candidates generation. In other words, for each query POI, the successive POIs to be recommended should have large total check-in count and meanwhile be located in its surrounding grids.
- **PMF+LR**: Probabilistic matrix factorization (PMF) is a standard approach for recommendation tasks [17], which can be extended by incorporating distance factor for traditional POI recommendation task [3]. We adapt this model to the successive POI recommendation task by additionally considering localized region

constraint. Thus, it can be treated as a direct adaptation of traditional POI recommendation approach to successive POI recommendation task.

- **FPMC+LR**: FPMC+LR is the state-of-the-art method for successive POI recommendation task [4]. Compared to PMF+LR, it utilizes the users' current visiting POI information.
- **LTSCR**: Location and time aware social collaborative retrieval model is the perfect model we proposed which integrates social relation, location information, temporal information with collaborative retrieval framework. We hope to verify its superiority.

Besides, it is also necessary to verify the effectiveness of the main components of LTSCR. To achieve this, we adopt the strategy similar to the forward selection method which is frequently employed in feature selection task [8]. Specifically, starting from the basic model (FPMC+LR), we add one more component that we intend to test each time and compare them in an incremental fashion. Here, we conclude the two partial methods to be adopted in later comparison as follows,

- **FPMC+LR+S**: The approach incorporates social relations into FPMC+LR. If the results show that this model outperforms FPMC+LR, then it can demonstrate users indeed prefer to visit POIs that their friends have visited empirically.
- **FPMC+LR+ST**: This model further considers temporal information of visiting. We adopt the same way as LTSCR used to model temporal information. If it behaves better than the previous method, the benefit of temporal information is experimentally verified.

Based on the results of the above partial methods of LTSCR, we can also demonstrate the advantage of utilizing the merits of CRM by comparing LTSCR with FPMC+LR+ST.

5.4 Parameter Settings

In this part, we mainly describe the hyper-parameters we tuned for LTSCR on the validation datasets. The other adopted comparison methods are also tuned to ensure to achieve their good results.

5.4.1 Learning Rate and Factor Dimension

There is a tradeoff between learning speed and convergence guarantee for learning rate setting. Without specific mention, we set learning rate η to be 0.01 for all models. The dimension of all latent factors are set to be 20 for Brightkite

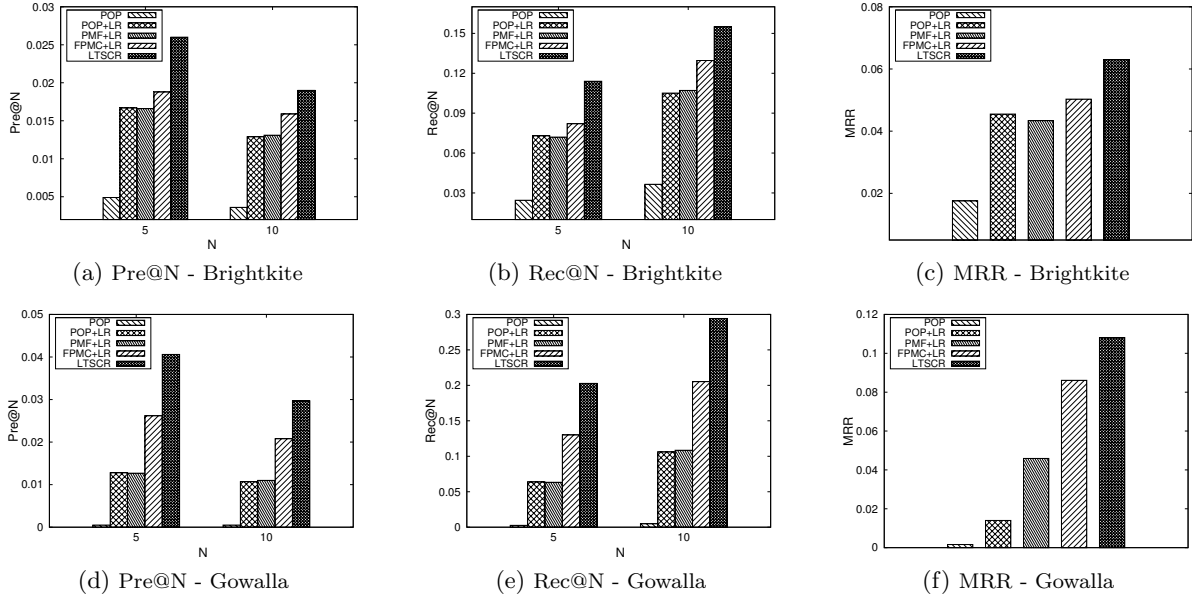


Figure 5: Comparisons of Different Methods on Recall@k, Precision@k, and MRR Metric.

dataset and 40 for Gowalla in our most experiments. This is inspired by the analysis of dimension influence on recommendation performance in Section 5.5.3.

5.4.2 Tuning of Parameter α and β

As shown in Equation 8, α and β control the relative influence of different parts on preference score computation. Here, we determine them based on the performances of Precision@5 and Recall@5 on validation datasets which coincide with the recommendation objective. Specifically, we employ the grid search method to get optimal setting by configuring α and β with different combinations. As Figure 4 shows, temporal factors of POIs play a minor role in optimal results compared with the other two parts. The optimal parameter configurations are pointed out in both figures and it is evident that the variations of results near the optimal point are relatively smooth. Without loss of generality, we set $\alpha = 0.2, \beta = 0.1$ for Brightkite data and $\alpha = 0.4, \beta = 0$ for Gowalla data, respectively.

5.4.3 Regularization Parameters

The regularization terms λ for all latent factors are set to be 0.1 across all the models based on the results on validation dataset after several trials on $\{10^{-3}, 10^{-2}, \dots, 10^3\}$. It is intuitive that γ and σ should be larger than λ as they capture additional knowledge to constrain the latent factors. We experimented many combinations for γ and σ . Experimentally, we find that the results continue to grow slowly with increase of γ and σ as long as they are larger than 1 and not too large. Empirically, we set them to be 10 which can ensure good results.

5.5 Experimental Results

5.5.1 Comparison Study

We start with discussing the results reported in Figure 5. Firstly, we compare the two POI popularity based methods, i.e., POP with POP+LR. From the table, we can see that POP+LR performs consistently better than POP in

both datasets. The phenomenon is rational since POP+LR utilizes the localization property of successive check-ins and ranks POIs only in local regions.

Then we analyze the results of the basic model based approach PMF+LR. Actually, we find the results of PMF+LR shows no improvements over POP+LR’s, except the result corresponding to MRR metric in Gowalla dataset. Knowledge gotten from the above comparison is two-folds. It reflects that PMF+LR is not very suited for the successive POI recommendation task. Besides, popularity based method is not very robust when the size of candidate POIs is large as their performance drops significantly in the large Gowalla dataset. By comparing FPMC+LR with the above methods, we can conclude that the incorporation of users’ current visiting locations is crucial for the successive POI recommendation task. This can be explained by that the current location each user stays in can be regarded as an implicit query which is an indicative for his successive check-in choice.

Finally, we contrast LTSCR with all other models. As the results show, LTSCR behaves vastly superior to both PMF+LR and FPMC+LR, which reveals that our proposed approach improves the state-of-the-art results significantly. The analysis of the key components of LTSCR is introduced subsequently. As we mentioned before, for each location and time aware query, the number of true recommendation results is small for successive POI recommendation task. Therefore, the precision of results may be relative low as long as n in Precision@ n is not small. However, the relative improvements on Precision@ n are significant. For example, LTSCR achieves 38.3% and 54.9% relative improvements over FPMC+LR on metric Precision@5 for datasets Brightkite and Gowalla, respectively.

5.5.2 Effectiveness of Main Components

We verify the effectiveness of utilizing social relation, temporal information, and collaborative retrieval framework in LTSCR through an incremental comparison strategy using the results shown in Figure 6.

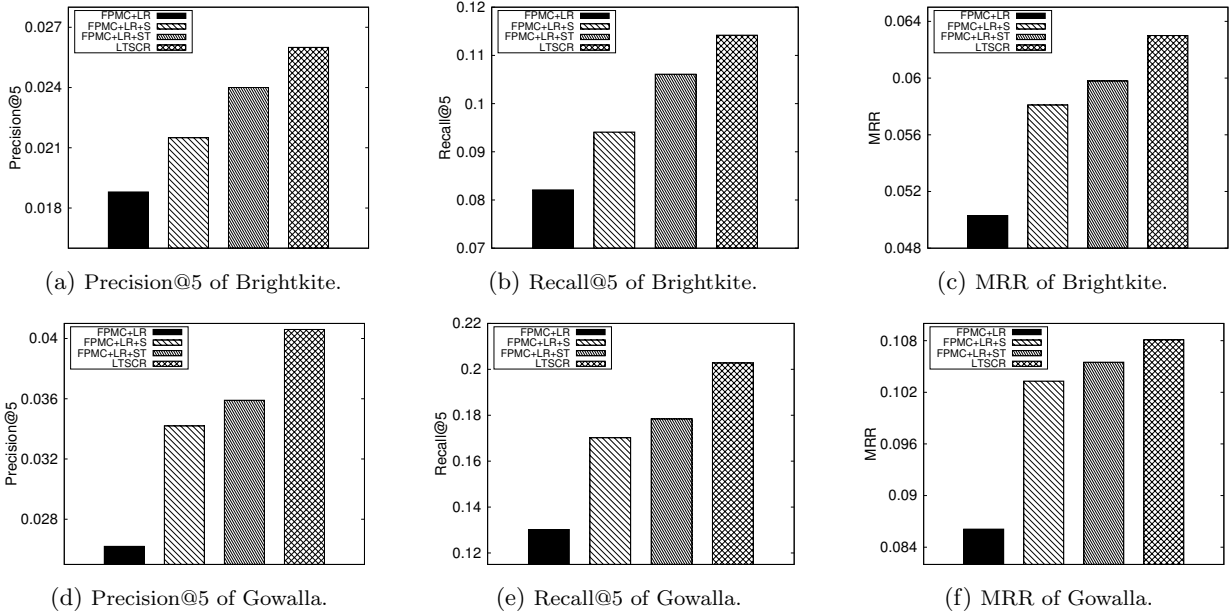


Figure 6: Performance of Different Components of LTSCR.

First, FPMC+LR+S consistently outperforms FPMC+LR in the two datasets notably. This indicates the effectiveness of using social relation for our task. Based on FPMC+LR+S, FPMC+LR+ST is further provided which is a direct extension of FPMC+LR+S by incorporating temporal information in the model, just as we mentioned in Section 5.3. The comparison between FPMC+LR+ST and FPMC+LR+S fits in with our expectation, as FPMC+LR+ST achieves better results. Thus, considering temporal information can benefit the task. Finally, to demonstrate the superiority of collaborative retrieval framework, we make a comparison between LTSCR and FPMC+LR+ST. As expected, LTSCR behaves the best among all the methods and outperforms FPMC+LR+ST.

In summary, through the above incremental comparisons, we can conclude that the main components we utilized are suited for the successive POI recommendation task.

5.5.3 Dimension Influence

In this section, we will briefly discuss how the dimension of latent factors influences the results. We conduct experiments on Precision@5 and Recall@5 on two datasets separately with different dimension setting changing from 5 to 100. The results are reported in Figure 8.

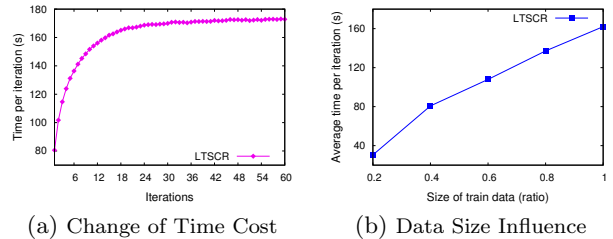
Clearly, the variation trend of the results on Brightkite dataset is not very stable. Although almost all the methods increase monotonously before reaching dimension 10, PMF+LR encounters an inflection point at this dimension. Purely increasing the dimension of latent factors may cause the overfitting problem, especially on the Brightkite dataset which is not very large. From a whole perspective, setting dimension to be 20 for Brightkite is an acceptable choice.

On the other hand, the curves drawn in Figure 8(c) and 8(d) are much clearer to reflect how the change of the dimension influences the results on Gowalla dataset. When the dimension is small, i.e., less than 30, the growth rate is fast. Then they start to converge after the dimension surpass 30 and retain pretty stable after 40.

From the two figures, we can also conclude that our proposed method consistently outperforms the other two model-based approaches, no matter how large the dimension is.

5.5.4 Learning Cost Analysis

Under the SGD learning framework, the total time of learning for LTSCR is mainly dominated by the cost of each learning iteration and the number of iterations.



(a) Change of Time Cost (b) Data Size Influence

Figure 7: Time Cost of LTSCR on Gowalla.

In each iteration, the time cost depends on the size of training instances and learning speed for each instance. As we introduced in Section 3.3, the sampling process for WARP loss could only be terminated when a sampled item satisfies the margin-based comparison. Thus, At the beginning of learning, the time cost for each instance is small as it is rapid to sample a negative item which satisfies the requirement. As the optimization process proceeds, the time cost will increase. Theoretically, the maximal number of sampling step for a training pair is $|\mathcal{P}| - 1$. However, when the optimization converges, the time cost of sampling will also be bounded. We tested the LTSCR’s learning time cost of each iteration on the large Gowalla dataset, which has nearly one million training pairs. As shown in Figure 7(a), when the time cost reaches a stable state, it is only slightly over 2 times as large as the time cost of the beginning, which is acceptable for model learning.

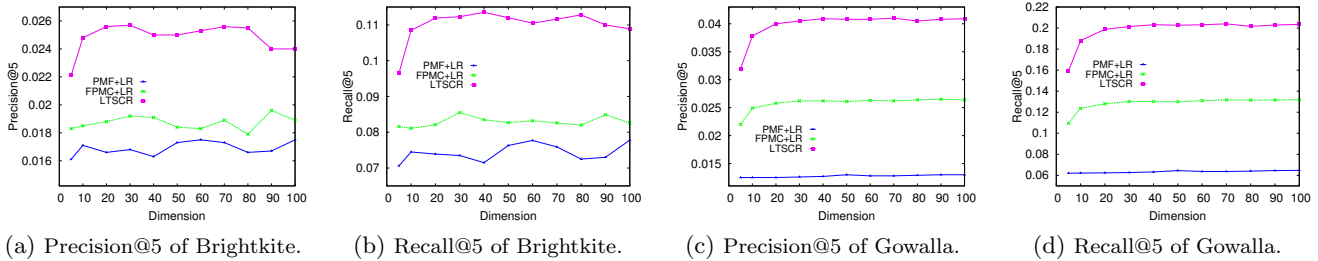


Figure 8: Results with Varying Dimension.

Besides, we tested how the data size affects the time cost of the learning of LTSCR on the large Gowalla dataset. From the results shown in Figure 7(b), we can estimate the average time cost of one iteration grows about linearly with the size of the dataset.

6. CONCLUSION

In this paper, we have proposed a new latent factor model called LTSCR for successive POI recommendation task. The new model not only considers users' current visiting locations, temporal information of visiting, social relations, and the localization of successive check-ins simultaneously, but also utilizes the merits of collaborative retrieval model. We validate the model on two real datasets. The experimental results have shown that 1) social and temporal information are indeed beneficial for this task, and 2) LTSCR outperforms the several other methods.

Acknowledgments

We thank the anonymous reviewers for their valuable and constructive comments. This work was supported in part by National Basic Research Program of China (973 Program) under Grant No. 2011CB302206, National Natural Science Foundation of China under Grant No. 61272088, and Tsinghua University Initiative Scientific Research Program.

7. REFERENCES

- [1] A. Ahmed, L. Hong, and A. J. Smola. Hierarchical geographical modeling of user locations from social media posts. In *WWW*, pages 25–36, 2013.
- [2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.
- [3] C. Cheng, H. Yang, I. King, and M. R. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *AAAI*, 2012.
- [4] C. Cheng, H. Yang, M. R. Lyu, and I. King. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, 2013.
- [5] H. Cheng, J. Ye, and Z. Zhu. What's your next move: User activity prediction in location-based social networks. In *SDM*, pages 171–179, 2013.
- [6] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *KDD*, pages 1082–1090, 2011.
- [7] H. Gao, J. Tang, X. Hu, and H. Liu. Exploring temporal effects for location recommendation on location-based social networks. In *RecSys*, pages 93–100, 2013.
- [8] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, Mar. 2003.
- [9] K.-J. Hsiao, A. Kulesza, and A. O. H. III. Social collaborative retrieval. In *WSDM*, pages 293–302, 2014.
- [10] B. Hu, M. Jamali, and M. Ester. Spatio-temporal topic modeling in mobile social media for location recommendation. In *ICDM*, pages 1073–1078, 2013.
- [11] T. Kurashima, T. Iwata, T. Hoshida, N. Takaya, and K. Fujimura. Geo topic model: joint modeling of user's activity area and interests for location recommendation. In *WSDM*, pages 375–384, 2013.
- [12] R. Li, S. Wang, H. Deng, R. Wang, and K. C.-C. Chang. Towards social user profiling: unified and discriminative influence model for inferring home locations. In *KDD*, pages 1023–1031, 2012.
- [13] B. Liu, Y. Fu, Z. Yao, and H. Xiong. Learning geographical preferences for point-of-interest recommendation. In *KDD*, pages 1043–1051, 2013.
- [14] X. Liu, Q. He, Y. Tian, W.-C. Lee, J. McPherson, and J. Han. Event-based social networks: linking the online and offline social worlds. In *KDD*, pages 1032–1040, 2012.
- [15] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *WSDM*, pages 287–296, 2011.
- [16] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, pages 811–820, 2010.
- [17] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, 2007.
- [18] J. Sang, T. Mei, J.-T. Sun, C. Xu, and S. Li. Probabilistic sequential pois recommendation via check-in data. In *SIGSPATIAL*, pages 402–405, 2012.
- [19] N. Usunier, D. Buffoni, and P. Gallinari. Ranking with ordered weighted pairwise classification. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, pages 1057–1064, New York, NY, USA, 2009. ACM.
- [20] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: Learning to rank with joint word-image embeddings. *Mach. Learn.*, 81(1):21–35, Oct. 2010.
- [21] J. Weston, C. Wang, R. J. Weiss, and A. Berenzweig. Latent collaborative retrieval. In *ICML*, 2012.
- [22] D.-N. Yang, C.-Y. Shen, W.-C. Lee, and M.-S. Chen. On socio-spatial group query for location-based social networks. In *KDD*, pages 949–957, 2012.
- [23] N. Yang, X. Kong, F. Wang, and P. S. Yu. When and where: Predicting human movements based on social spatial-temporal events. In *SDM*, pages 515–523, 2014.
- [24] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, pages 325–334, 2011.
- [25] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen. Lcars: A location-content-aware recommender system. In *KDD*, pages 221–229, New York, NY, USA, 2013. ACM.
- [26] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. Magnenat-Thalmann. Time-aware point-of-interest recommendation. In *SIGIR*, pages 363–372, 2013.
- [27] W. Zhang and J. Wang. A collective bayesian poisson factorization model for cold-start local event recommendation. In *KDD*, Sydney, Australia, 2015.

- [28] W. Zhang, J. Wang, and W. Feng. Combining latent factor model with location features for event-based group recommendation. In *KDD*, pages 910–918, Chicago, IL, USA, 2013.