Personalized Location Recommendations with Local Feature Awareness

Xiaoyan Zhu*, Ripei Hao*, Haotian Chi*, Xiaojiang Du† *National Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China
†Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA
Email: xyzhu@mail.xidian.edu.cn, haoripei@163.com, htchi@stu.xidian.edu.cn, dux@temple.edu

***Abstract*—Location-based social networks (LBSNs) make it possible for people to record their location histories, mine their life patterns, and infer individual preferences. As an important component of LBSNs, recommender systems gained popularity in recent years. Recommender systems can automatically list candidate locations for users according to their preference; this is different from traditional search methods, which finds locations that users may prefer. However, making effective recommendations suffers from data sparsity. To relieve this problem and achieve high effectiveness, we take context information into consideration and present a personalized location recommender system considering both user preference and local features in this paper. To be specific, we apply Labeled-LDA in user preference learning and local features inference processes, which are denoted as UL-LDA model and CL-LDA model, respectively. Because of this, we can make recommendations even on the condition that users are in a new city and have little information about the city. We evaluate our approach with extensive experiments on a large-scale Foursquare dataset. The experimental results clearly validate the effectiveness of our approach.**

I. INTRODUCTION

Location-based social networks (LBSNs), like Foursquare, as one kind of online social networks (OSNs), allows users to share their location by *check-in* via a smartphone or SMS accompanied with location-acquisition technologies. Advances in broadband wireless networks and location sensing technologies, which allow ubiquitous access to the web through smart mobile phones, tablets and so on, lead to the wide spread of LBSN. As of December 2013, Foursquare had 45 million registered users. Additionally, the company recorded 382 million *check-ins* from its 7 million members in 2010. Huge amount of records of *who* visited *where* in *which city* imply extensive knowledge about an individual's preference. By taking into account the places that users visit and the things they have told the service provider that they like, we can provide recommendations of locations for users, which improves satisfaction and reduces the time and energy costs.

Recommending locations is necessary and valuable for both users and service providers. On one hand, reasonable recommendations provide suitable locations or other services for targeted users, which reduces the time and energy that is required for searching user locations. It is especially important when users travel to unfamiliar areas. For example, a user living in Manhattan may go to Brooklyn for the first time, which means no location in Brooklyn has been visited by the user. Recommending proper locations can help them save energy and enjoy their stay. On the other hand, recommendation helps to increase benefits of the service provider.

Since location recommendation is so useful in LBSN, how can we achieve high quality recommendations that satisfy the targeted user's preference, especially in sparse data situations? Traditional item-based and user-based collaborative filtering [1], [2], [3] perform poorly in sparse data situations. [4], [5], [6] exploit trust relations in recommendation algorithms. Moreover, experiments of these schemes demonstrate that trust-based recommendation increases recommendation accuracy and can effectively deal with the cold-start user problem. Bao et al. [7] presented a location-based and preference-aware recommender system that offered a particular user a set of locations within a geospatial range with the consideration of user personal preferences and social opinions. They use weighted category hierarchy to model personal preference and then learn the expertise of each user in a city, which is later used to find local experts. Finally, by combining the opinions of local experts and personal preferences, personalized recommendations are generated. In our scheme, we take local features and location category into account. Location context plays an important role in recommendation; this has been investigated in [8]. Interests of people in different cities vary largely. Interest localization can be vividly suggested from the report *A Peek into Netflix Queue* in New York Times. To achieve local feature aware personalized location recommendations, we need to first of all solve three major challenges: 1) how can we infer user preference? 2) how can we infer local features? 3) how can we combine user preference and local features in an effective way?

An improved Latent Dirichlet Allocation model Labeled-LDA is introduced. To be specific, we build a User-based Labeled-LDA (UL-LDA) model to mine an individual's preference using user profiles. In this model, topics are restricted to location categories in user profiles. Similar to user preference mining, we construct a City-based Labeled-LDA (CL-LDA) model to infer local features based on location profiles. Topics in CL-LDA are restricted to location categories in city profiles. By capturing local folk-customs, recommendations are more convincing. At last, the tradeoff between user preferences and local features is balanced by adding proper weights to these two factors. On one hand, user preference is a necessary component that we need to fully considered in the recommender system, because the system is aimed to provide locations that satisfy personal taste. On the other hand, local features cannot be ignored in location recommendations, especially on the condition that little user information is available. We use Relative Standard Deviation (RSD) to learn the weights of the two factors. In summary, we propose a location recommender system that offers a particular user a set of locations within a certain city with local feature awareness in this paper. Our contributions are listed as follows:

- We adopt UL-LDA and CL-LDA models to learn user preference and local feature, respectively. UL-LDA and CL-LDA models can mimic human decision making on locations.
- We use Relative Standard Deviation to learn the weights of user preference and local feature based on their importance learned from dataset. Differing from the arithmetic mean, the weighted mean is more accurate.
- We evaluate our recommendation system using a real-world dataset collected from Foursquare. The experiment shows that our system is effective.

The rest of this paper is organized as follows: Section II lists related work. Section III presents the overview of our system. Section IV describe our scheme in detail. Section V performs experiments. Finally, Section VI concludes the paper.

II. Related Work

In this section, we present existing literature on location recommendation, ranging from generic location recommendations to personalized location recommendation and traditional collaborative filtering recommendation algorithms to the state-of-the-art recommendation algorithms.

A generic location recommendation system generates the most popular locations to users without considering their personal preferences. Zheng et al. [9] performed generic recommendations that provided users with the top interesting locations and travel sequences. In the scheme, they first used TBHG to model users' histories and then adopted an HITS-based model to infer hot locations and experts.

Personalized recommendation has been widely studied. C-ityVoyager [10] had been designed to recommend shops based on users' past location data history by using an item-based collaborative filtering algorithm. Zheng et al. [11] proposed a scheme that used a collaborative filtering (CF) model to conduct a personalized location recommendations. To be specific, in order to make recommendation, they used the similarity between each pair of users, which offered locations matching an individual's preferences. Additionally, Zheng et al. [12] proposed a solution to mine interesting locations and classical travel sequences in a given geospatial region based on multiple users' GPS trajectories. The key insight is that users' travel experiences and location interests have a mutual reinforcement relationship. In the paper, they took into account both the sequence property of people's movement behaviors and the hierarchy property of geographic spaces. A framework, referred to as hierarchical-graph-based similarity measurement (HGSM), was proposed for geographic information systems to consistently model each individual's location history and effectively measure the similarity among users. Zheng et al. [9] also proposed a personalized recommendation system. They applied an HITS-based model to learn popular spots from experienced users and then adopt a user-based collaborative filtering algorithm to make recommendations.

However, pure collaborative filtering approaches [1], [2], [3] are challenged with data sparsity and the cold-start problem. To overcome the problem, different methods have been proposed. Ma et al. [13] came up with a probability matrix factorization framework, which fused users' tastes and their friends' favors, to produce recommendations. LARS* [14] is a location-aware recommender system that uses location-based ratings to produce recommendations. Bao et al. [7] presented a recommender system that offered a particular user a set of locations (such as restaurants and shopping malls) within a geospatial range taking the user's personal preferences and social opinions into consideration. This recommender system can facilitate people to travel not only near their living areas but also to a city that is new to them.

III. System Overview

In this section, we first present basic concepts that we use in our scheme, and then take a glance at the overall architecture of the proposed location recommender system.

A. Preliminary

In a location-based social network (take Foursquare, for example) registered users use *check-ins* to mark locations that they visited with some comments. Each location is labeled with its category and a pair of coordinates inferring its geographical position. In Foursquare, locations are divided into ten coarse-grained categories C={Arts & Entertainment, College & University, Event, Food, Nightlife Spot, Outdoor & Recreation, Professional & Other places, Residence, Shop & Service, Travel & Transport}

TABLE I
Segment of one user profile

User ID	Location	City	Location Category
2	Pasadena City College	Pasadena	College & University
2	Westfield Santa Anita	Arcadia	Shop & Service
2	Garden of the Dragon	Las Vegas	Food

Definition 1. (User Profile) The profile of user u contains all records of the user and each record is a tuple (u, v, l_v, c_v), meaning that a user u visited a location v in the city l_v and the location v belongs to category c_v, where $u \in U = \{u_1, , \ldots, u_N\}$, $v \in V = \{v_1, , \ldots, v_P\}$, $l \in L = \{l_1, , \ldots, l_M\}$ and $c \in C$. Note that a location may correspond to more than one category. For example, a park generally belongs to $Outdoor\&Recreation$ category while sometimes events are held there. A segment of one user' profile is shown in Table I.

TABLE II
Segment of one city profile

City	User ID	Location	Location Category
New York	1	Exit Art	Arts & Entertainment
New York	5	Rattle N Hum	Nightlife Spot
New York	13	Bowery Ballroom	Arts & Entertainment

Definition 2. (City Profile) A city profile is a simple transformation of the user profile. The city profile contains all locations that different users visited in the city. Similar to a user profile, each record in a location profile is a tuple (l, u, v, c_v) inferring that in location l, a user u visited a location v belonging to category c_v, as shown in Table II, where $u \in U = \{u_1, , \ldots, u_N\}$, $v \in V = \{v_1, , \ldots, v_P\}$, $l \in L = \{l_1, , \ldots, l_M\}$ and $c \in C$.

Definition 3. (Labeled-LDA) Labeled-LDA [15] is a probabilistic generative model, which was originally constructed to describe the process of generating a labeled document collection. Besides modeling each profile as a mixture of underlying topics and generating each location from one topic, Labeled-LDA incorporates supervision by constraining topics to a location category label set.

B. System Architecture

Our system mainly consists of three parts: user preference mining, local feature inference, and top-k recommendation, as shown in Fig.1.

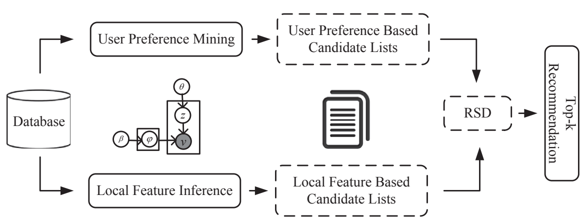


Fig. 1. System Architecture.

User Preference Mining. Given user profiles, we apply the UL-LDA model to infer an individual's preference. Then, we calculate ranking scores corresponding to users' preferences for locations, and then store the top-k locations in a list.

Local Feature Inference. For the local feature inference component, we model the features of a city, which is the common preference of people living in this area, using CL-LDA. Then, we calculate the top-k locations in this area and store them in order in a list.

Top-k Recommendation. After learning an individual's preferences and local features of a city, the top-k recommendation algorithm offers a list of locations in a particular city for a certain user. We take both user preferences and local features into account and add different weights to the two factors during recommendation. Weight represents the importance of the two factors when a user u makes a decision on which location to select. We use the Relative Standard Deviation algorithm to compute these weights. Then, we construct an unified equation to estimate the rating that the user would give to an unvisited candidate location. Later, the top-k locations with relatively high prediction ratings are returned as recommendations.

IV. System Introduction

User preference mining and local feature inference can be conducted offline. The user preference mining part learns a user's preferences and the local feature inference part evaluates a city's feature. Top-k recommendation is carried online after the user preference mining and local feature inference stages. We define the main notations used in this part in Table III for the ease of understanding and afterward presentation.

TABLE III
Notations used in models

NOTATIONS		DESCRIPTION
UL-LDA	LL-LDA	
α	α'	Dirichlet topic prior
β	β'	Dirichlet location prior
ϕ	ϕ'	label prior for topic
Λ	Λ'	binary topic presence/absence indicators
θ	θ'	parameters of the multinomial distribution corresponding to topics
φ	φ'	parameters of the multinomial distribution corresponding to locations (the former) and cities (the latter)

A. User Preference Mining

We deploy the UL-LDA model to infer a user's preference. At first, we define a list of binary topic presence/absence indicators $\Lambda^{(u)} = \{s_1, s_2, \ldots, s_K\}$, where K is the total number of unique labels in the corpus. We assume the number of topics in UL-LDA to be the number of unique categories of locations in users profiles. The preference of a user u for certain location v is sampled from the UL-LDA model, which is formulated as Equation 1:

$$P\left(v|\theta, \varphi\right) = \sum_{z} P\left(v|z, \varphi_z\right) P\left(z|\alpha^{(u)}\right) \quad (1)$$

where $\alpha^{(u)}$ are the restricted parameters of the Dirichlet topic prior. The generative process is shown in Algorithm 1. The procedure contains four phases, and its graphical model is shown in Fig. 2.

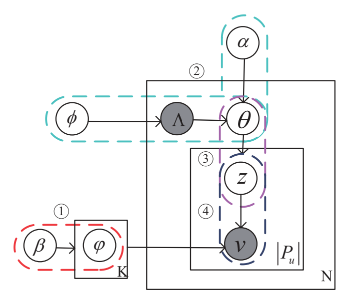


Fig. 2. Graphical model of UL-LDA.

1) $\beta \rightarrow \varphi$: drawing the multinomial topic distributions over locations φ for each topic k from a Dirichlet prior β, where $kinK$.
2) $\alpha \rightarrow \theta_u$: drawing a multinomial mixture distribution $\theta_{(u)}$ over restricted topics for each user profile from a Dirichlet prior α. The topic assignments are limited to the user profile's labels.

   The procedure [15] to restrict θ_u to be defined only over the topics that correspond to its labels $\Lambda^{(u)}$ is described in the following:

Algorithm 1 : Generative process for UL-LDA

for each topic $k \in \{1, \ldots, \mathrm{K}\}$ **do**
  draw $\varphi_k \sim Dir\left(\cdot|\beta\right)$;
end for
for each user profile p_u **do**
  for each topic $k \in \{1, \ldots, \mathrm{K}\}$ **do**
    generate $\Lambda^u \in \{0, 1\} \sim Bernoulli\left(\cdot|\phi_k\right)$;
  end for
  generate $\alpha^{(u)} = L^{(u)} \times \alpha$;
  generate $\theta^{(u)} \sim Dir\left(\cdot|\alpha^{(u)}\right)$;
  for each i in $\{1, \ldots, \mathrm{N}\}$ **do**
    draw $z_i \in \left\{\lambda_1^{(u)}, \lambda_2^{(u)}, \ldots, \lambda_{R_u}^{(u)}\right\} \sim Multi\left(\cdot|\theta^{(u)}\right)$;
    draw $v_i \in V \sim Multi\left(\cdot|\varphi_{z_i}\right)$;
  end for
end for

- generate the user profile's labels $\Lambda^{(u)}$ using a Bernolli coin toss for each topic k, with a labeling prior probability ϕ_k. Suppose there are six category labels. For each Bernoulli test, the result is zero or one, that is, $s_k = 0$ or $s_k = 1$. We take $\Lambda^{(u)} = \{s_1, s_2, \ldots, s_6\} = \{0, 0, 1, 0, 1, 1\}$ for example.
- define the vector of user profile's labels to be $\lambda^{(u)} = \left\{k|\Lambda_k^{(u)}\right\} = \{3, 5, 6\}$.
- define a user profile-specific label projection matrix $L^{(u)}$ whose size is $R_u \times K$ for each user profile as follows, where $R_u = \left|\lambda^{(u)}\right|$.

$$L_{ij}^{(u)} = \begin{cases} 1 & if\ \lambda_i^{(u)} = j \\ 0 & otherwise \end{cases}$$

Then the $L^{(u)}$ would be

$$L^{(u)} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- project the parameter vector of the Dirichlet topic prior $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_K)^T$ to a lower dimensional vector $\alpha^{(u)}$ using the $L^{(u)}$ as follows: $\alpha^{(u)} = L^{(u)} \times \alpha = \left(\alpha_{\lambda_1^{(u)}}, \alpha_{\lambda_2^{(u)}}, \ldots, \alpha_{\lambda_{R_u}^{(u)}}\right)^T$. In the example,

$$\alpha^{(u)} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{bmatrix} = \begin{bmatrix} \alpha_3 \\ \alpha_5 \\ \alpha_6 \end{bmatrix}$$

- draw $\theta^{(u)}$ from a Dirichlet distribution with parameters $\alpha^{(u)}$.

3) $\theta_u \rightarrow z$: generating the topic z from a multinomial mixture distribution θ_u.
4) $\varphi \rightarrow v$: generating location v from the multinomial distribution φ.

Following the studies [16], we use collapsed Gibbs sampling to obtain samples of the hidden variable assignments $P\left(z|v\right)$. In the sampling procedure, we begin with the joint probability $P\left(z, v\right)$ of all user profiles in our dataset. The posterior distribution of $P\left(\theta_u|z_{\neg i}, v_{\neg i}\right)$ and $P\left(\varphi_k|z_{\neg i}, v_{\neg i}\right)$ is.

$$P\left(\theta_u|z_{\neg i}, v_{\neg i}\right) = Dir\left(\theta_u|n_{u,\neg i} + \alpha^{(u)}\right) \tag{2}$$

$$P\left(\varphi_k|z_{\neg i}, v_{\neg i}\right) = Dir\left(\varphi_k|n_{k,\neg i} + \beta\right) \tag{3}$$

Then, the inference of Gibbs sampling function is:

$$\begin{aligned} & P\left(z_i = k|z_{\neg i}, v\right) \propto P\left(z_i = k, v_i = v|z_{\neg i}, v_{\neg i}\right) \\ & = \int p\left(z_i = k, \theta_u|z_{\neg i}, v_{\neg i}\right) p\left(v_i = v, \varphi_k|z_{\neg i}, v_{\neg i}\right) d\theta_u d\varphi_k \\ & = E\left(\theta_{uk}\right) E\left(\varphi_{kv}\right) \\ & = \hat{\theta}_{uk} \cdot \hat{\varphi}_{kv} \end{aligned}$$

After a sufficient number of sampling iterations, we can estimate the parameters θ_{uk} and φ_{kv}:

$$\hat{\theta}_{uk} = \frac{n_{u,\neg i}^{(k)} + \alpha_k^{(u)}}{\sum\limits_{k=1}^{R_u} \left(n_{u,\neg i}^{(k)} + \alpha_k^{(u)}\right)} \tag{4}$$

$$\hat{\varphi}_{kv} = \frac{n_{k,\neg i}^{(v)} + \beta_v}{\sum\limits_{v=1}^{|V|} \left(n_{k,\neg i}^{(v)} + \beta_v\right)} \tag{5}$$

Where $n_{u,\neg i}^{(k)}$ is the count of topic k sampled from user u's profile, that excludes the current assignment z_i. $n_{k,\neg i}^{(v)}$ is the count of location v sampled from the topic k, that excludes the current assignment z_i.

B. Local Feature Inference

The local feature inference process is similar to that of the user's preferences. Here, we omit the details and directly give the graphical model of CL-LDA, which is shown in Fig. 3. Then, the preference of a city l for location v is sampled from the model CL-LDA, which is formulated as Equation 6. Its generative process is described in Algorithm 2.

$$P\left(v|\theta'_l, \varphi'\right) = \sum_z P\left(v|z, \varphi'_z\right) P\left(z|\theta'_l\right) \tag{6}$$

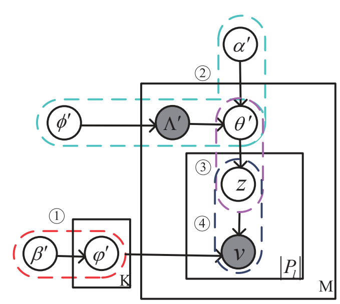


Fig. 3. Graphical model of CL-LDA

Based on city profiles, we use collapsed Gibbs sampling to obtain the hidden variable assignment $P\left(z_i = k|z_{\neg i}, v\right)$ and to estimate parameters θ' and φ'. The detail computation process is similar to that of UL-LDA. Then, we present the results as follows:

$$\hat{\theta}'_{lk} = \frac{n^{(k)}_{l,\neg i} + \alpha'^{(l)}_k}{\sum_{k=1}^{R_l} \left(n^{(k)}_{l,\neg i} + \alpha'^{(l)}_k \right)} \tag{7}$$

$$\hat{\varphi}'_{kv} = \frac{n^{(v)}_{k,\neg i} + \beta'_v}{\sum_{v=1}^{|V|} \left(n^{(v)}_{k,\neg i} + \beta'_v \right)} \tag{8}$$

$$P\left(z_i = k | z_{\neg i}, v\right) \propto \hat{\theta}'_{lk} \cdot \hat{\varphi}'_{kv} \tag{9}$$

Algorithm 2 : Generative process for CL-LDA

for each topic $k \in \{1, \ldots, \mathrm{K}\}$ **do**
  draw $\varphi'_k \sim Dir\left(\cdot|\beta'\right)$;
end for
for each location profile p_l **do**
  for each topic $k \in \{1, \ldots, \mathrm{K}\}$ **do**
    generate $\Lambda^u \in \{0, 1\} \sim Bernoulli\left(\cdot|\phi'_k\right)$;
  end for
  generate $\alpha'^{(l)} = L^{(l)} \times \alpha$;
  generate $\theta'^{(l)} \sim Dir\left(\cdot|\alpha'^{(l)}\right)$;
  for each i in $\{1, \ldots, \mathrm{M}\}$ **do**
    draw $z_i \in \left\{\lambda^{(l)}_1, \lambda^{(l)}_2, \ldots, \lambda^{(l)}_{R_l}\right\} \sim Multi\left(\cdot|\theta'^{(l)}\right)$;
    draw $v_i \in V \sim Multi\left(\cdot|\varphi'_{z_i}\right)$;
  end for
end for

C. Top-k Recommendation

The top-k recommendation can calculate the preferences of a user u for a certain location v, equipped with parameters θ, θ', φ and φ' evaluated from UL-LDA and CL-LDA models in the offline learning part. The ranking score of user u for a location v is computed as:

$$f_{uv} = \hat{\theta}_{uz} \hat{\varphi}_{zv}$$

The ranking score of the common attitude in the city l for a location v are computed separately as follows:

$$f'_{lv} = \hat{\theta}'_{lz} \hat{\varphi}'_{zv}$$

After obtaining f_{uv} and f'_{lv}, we then compute the weight of each factor. In our work, we consider user preferences and local features unequally and add weights to the two factors separately. To be specific, we use the Relative Standard Deviation to evaluate the importance that individual preferences and local features plays in a user u's decision-making process. Let ω_u denote the weight of user preferences and ω_l denote the weight of local features.

$$\omega_u = \frac{1}{|V_{p_u}| - 1} \frac{\sqrt{\sum_{v \in V_{p_u}} \sum_{z \in Z_{p_u}} \left(\hat{\theta}_{uz} \hat{\varphi}_{zv} - \overline{f} \right)^2}}{\frac{1}{|V_{p_u}|} \sum_{v \in V_{p_u}} \sum_{z \in Z_{p_u}} \hat{\theta}_{uz} \hat{\varphi}_{zv}} \tag{10}$$

$$\omega_l = \frac{1}{|V_{p_l}| - 1} \frac{\sqrt{\sum_{v \in V_{p_l}} \sum_{z \in Z_{p_l}} \left(\hat{\theta}'_{lz} \hat{\varphi}'_{zv} - \overline{f'} \right)^2}}{\frac{1}{|V_{p_l}|} \sum_{v \in V_{p_l}} \sum_{z \in Z_{p_l}} \hat{\theta}'_{lz} \hat{\varphi}'_{zv}} \tag{11}$$

Where V_{p_u} means locations contained in the user u's profile p_u and V_{p_l} means locations contained in the city l's profile p_l. After computing the weights of user preferences and local features, we can then normalize them as Equation 12. Thus, the prediction score of a candidate location that the user u gives is evaluated in Equation 13.

$$\lambda_u = \frac{\omega_u}{\omega_u + \omega_l}, \lambda_l = \frac{\omega_l}{\omega_u + \omega_l} \tag{12}$$

$$R\left(u, v\right) = \lambda_u f_{uv} + \lambda_l f'_{lv} \tag{13}$$

where $\lambda_u + \lambda_l = 1$. Later, the top-k locations with relative high predict ratings are returned as recommendations.

V. Experiment Evaluation

This section contains two subsections: the first one is about experiment settings-**dataset, baselines, and measurements**; the other is experiment result, showing effectiveness of our scheme.

A. Experiment Setting

Dataset We use a publicly available dataset of a location-based social network, Foursquare, in our experiment [17]. The dataset contains 4163 users and 483709 records. The following information is recorded when collecting the data:

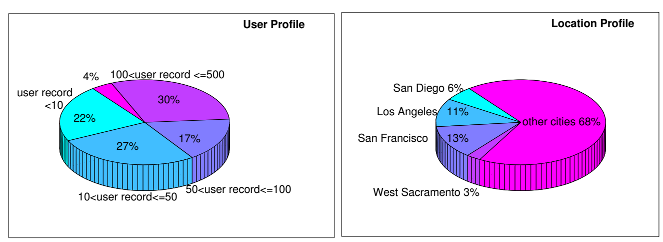


Fig. 4. Demographic statistics of our dataset. The left illustrates an analysis of user profiles and the right depicts an analysis of city profiles.

Baselines We compare our method with the following three baseline approaches.

- *Location-based and preference-aware recommendation (LPA)* This scheme offers location recommendations for a particular user considering both user preferences and social opinions [7].
- *User preference aware rcommendation (UPAR)* As one part of our recommendation scheme, user preference aware recommendation only considers personal interests.
- *Local feature aware recommendation (LFAR)* As the other part of our recommendation scheme, local feature aware recommendation takes folk-custom into consideration instead of individual preference.

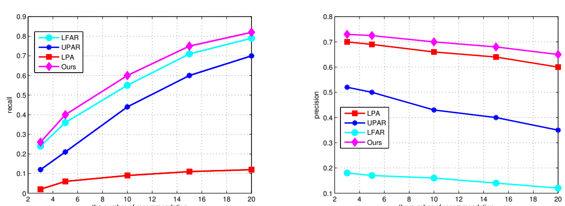
Measurements To evaluate the effectiveness of our system, we divide a user's location history into two parts: 1) we select the location history generated in a querying city as a test set; 2) we use the rest of the user's location history as the training set.

We regard the locations that a user has visited in a queried city as the ground truths and match the recommended locations against them. The more recommended locations visited by a user in the test city, the more effective the recommendation method is. Based on the ground truth and recommendations, we are able to compute two kinds of measurements: precision and recall, which is shown in the following. The higher the precision is, the better the recommendation scheme is. It is the same to recall.

$$recall = \frac{number\, of\, recovered\, ground\, truths}{total\, number\, of\, ground\, truths}$$

$$precision = \frac{number\, of\, recovered\, ground\, truths}{total\, number\, of\, recommendations}$$

B. Experiment result



(a) Recall varies with the number of recommendations.

(b) Precision varies with the number of recommendations.

Fig. 5. Precision and Recall Comparison of different methods.

Figure 5(a) and Figure 5(b) illustrate the average recall and precision of different methods varying with the number of recommendations. The figures show that both the precision and recall measurements are higher than that of LPA, suggesting that the performance of our method is superior to LPA. At the same time, our method and LPA outperform LFAR and UPAR, which demonstrates the importance of individual preferences and local features, respectively. Another observation is that UPAR outperforms LFAR in our setting, which verifies the benefit brought by personalized recommendation.

Figure 6 shows that the performance of our method is stable in sparse data situation. UPAR is greatly influenced by the data density. When the data density is very low, the average precision of UPAR drops sharply. In contrast, the average precision of LFAR remains stable, suggesting the importance of local features in recommendations when the data is sparse. To summarize, the experiment results validate the effectiveness of our scheme.

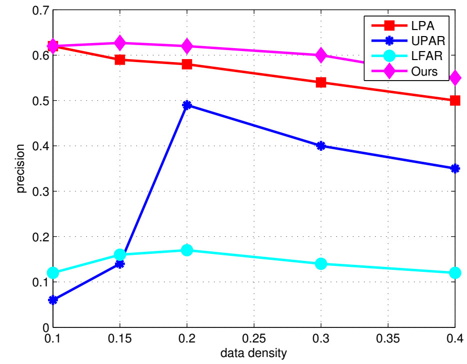


Fig. 6. Precision varies with data density.

VI. Conclusion

In this paper, we proposed a location-preference-aware recommender system to provide locations for users. It considers both user preferences and local features to make recommendations. Moreover, by considering category labels tagged to locations, we can alleviate the data sparsity and improve the precision of recommendations. The recommender system works when you are in an unfamiliar area. We evaluated our system using real-world dataset and the results show the effectiveness of our scheme.

References

[1] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999, pp. 230–237.

[2] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001, pp. 285–295.

[3] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 143–177, 2004.

[4] M. Jamali and M. Ester, "Trustwalker: a random walk model for combining trust-based and item-based recommendation," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 397–406.

[5] ——, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 135–142.

[6] R. Forsati, I. Barjasteh, F. Masrour, A.-H. Esfahanian, and H. Radha, "Pushtrust: An efficient recommendation algorithm by leveraging trust and distrust relations," in *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 2015, pp. 51–58.

[7] J. Bao, Y. Zheng, and M. F. Mokbel, "Location-based and preference-aware recommendation using sparse geo-social networking data," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, 2012, pp. 199–208.

[8] T. Horozov, N. Narasimhan, and V. Vasudevan, "Using location for personalized poi recommendations in mobile environments," in *Applications and the Internet, 2006. SAINT 2006. International Symposium on*. IEEE, 2006, pp. 6–pp.

[9] Y. Zheng and X. Xie, "Learning travel recommendations from user-generated gps traces," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 1, p. 2, 2011.

[10] Y. Takeuchi and M. Sugimoto, "Cityvoyager: an outdoor recommendation system based on user location history," in *Ubiquitous intelligence and computing*. Springer, 2006, pp. 625–636.

[11] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma, "Recommending friends and locations based on individual location history," *ACM Transactions on the Web (TWEB)*, vol. 5, no. 1, p. 5, 2011.

[12] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," ACM, pp. 791–800, 2009.

[13] H. Ma, I. King, and M. R. Lyu, “Learning to recommend with social trust ensemble,” in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2009, pp. 203–210.

[14] M. Sarwat, J. J. Levandoski, A. Eldawy, and M. F. Mokbel, “Lars*: An efficient and scalable location-aware recommender system,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 6, pp. 1384–1399, 2014.

[15] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, “Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora,” in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 248–256.

[16] T. L. Griffiths and M. Steyvers, “Finding scientific topics,” *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.

[17] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen, “Lcars: a location-content-aware recommender system,” in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 221–229.