LCARS: A Location-Content-Aware Recommender System

Hongzhi Yin[†] Yizhou Sun[§] Bin Cui[†] Zhiting Hu[†] Ling Chen[‡]
[†]Department of Computer Science and Technology
Key Laboratory of High Confidence Software Technologies, Peking University
[§]College of Computer and Information Science, Northeastern University
[‡]QCIS, University of Technology, Sydney
[†]{bestzhi, bin.cui, huzhiting}@pku.edu.cn, [§]yzsun@ccs.neu.edu, [‡]ling.chen@uts.edu.au

ABSTRACT

Newly emerging location-based and event-based social network services provide us with a new platform to understand users' preferences based on their activity history. A user can only visit a limited number of venues/events and most of them are within a limited distance range, so the user-item matrix is very sparse, which creates a big challenge for traditional collaborative filtering-based recommender systems. The problem becomes more challenging when people travel to a new city where they have no activity history.

In this paper, we propose LCARS, a location-content-aware recommender system that offers a particular user a set of venues (e.g., restaurants) or events (e.g., concerts and exhibitions) by giving consideration to both personal interest and local preference. This recommender system can facilitate people's travel not only near the area in which they live, but also in a city that is new to them. Specifically, LCARS consists of two components: offline modeling and online recommendation. The offline modeling part, called LCA-LDA, is designed to learn the interest of each individual user and the local preference of each individual city by capturing item co-occurrence patterns and exploiting item contents. The online recommendation part automatically combines the learnt interest of the querying user and the local preference of the querying city to produce the top-k recommendations. To speed up this online process, a scalable query processing technique is developed by extending the classic Threshold Algorithm (TA). We evaluate the performance of our recommender system on two large-scale real data sets, DoubanEvent and Foursquare. The results show the superiority of LCARS in recommending spatial items for users, especially when traveling to new cities, in terms of both effectiveness and efficiency.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering; J.4 [**Computer Applications**]: Social and Behavior Sciences

Keywords

Recommender system; Location-based service; Probabilistic generative model; TA algorithm; Cold start

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.

1. INTRODUCTION

Newly emerging event-based social network services (EBSNs), such as Meetup (`www.meetup.com`) and DoubanEvent (`www.douban.com/events/`), provide online platforms for users to establish social events which will be held in physical places [16]. Given a created social event, users may express their intent to join by replying "yes", "no" or "maybe". Meanwhile, the advances in location-acquisition and wireless communication technologies enable users to add a location dimension to traditional social networks, fostering the growth of location-based social networking services (LBSNs), such as Foursquare (`foursquare.com`) and Gowalla (`gowalla.com`) which allow users to "check-in" at spatial *venues* and rate their visit via mobile devices.

In this paper, we aim to mine more knowledge from the user activity history data in LBSNs and EBSNs to answer two typical types of questions that we often ask in our daily: (1) If we want to visit venues in a city such as Beijing, where should we go? (2) If we want to attend local events such as dramas and exhibitions in a city, which events should we attend? In general, the first question corresponds to venue recommendations, and the second question corresponds to event recommendations. By answering these two questions, we can satisfy the personalized information needs for many users in their daily routines and trip planning. For simplicity, we propose the notion of *spatial items* to denote both venues and events in a unified way, so that we can define our problem as follows: given a querying user u with a querying city l_u, find k interesting spatial items within l_u, that match the preference of u.

However, inferring user preferences for spatial items is very challenging by using users' activity history in an EBSN or LBSN. First, a user can only visit a limited number of physical venues and attend a limited number of social events. This leads to a sparse user-item matrix for most existing location-based recommender systems [14, 10], which directly use collaborative filtering-based methods [20] over spatial items. Second, the observation of travel locality [14] makes the task more challenging considering that a user travels to a new place where he/she does not have any activity history. The observation of travel locality made on EBSNs and LBSNs shows that users tend to travel a limited distance when visiting venues and attending events. An investigation shows that the activity records generated by users in their non-home cities are very few and only take up 0.47% of the activity records they left in their home cities. This observation of travel locality is quite common in the real world [22], aggravating the data sparsity problem with personalized spatial item recommendations (e.g., if we want to suggest spatial items located in Los Angeles to people from New York City). In this case, solely using a CF-based method is not feasible any more, especially when coping with the *new city* problem, be-

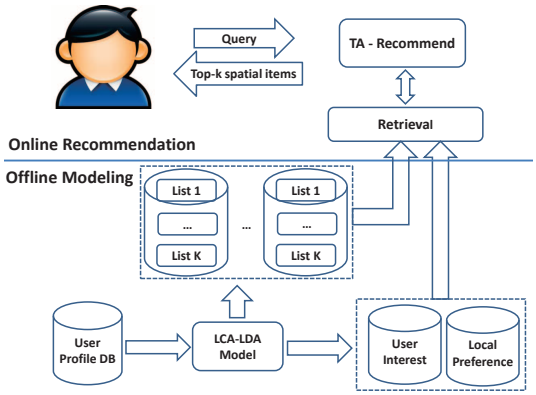


Figure 1: The Architecture Framework of LCARS

cause a querying user usually does not have enough activity history of spatial items in a city that is new to him/her.

Let us assume, for example, that querying user u is a Shopaholic and often visits shopping mall v' in his/her home city; v is a popular local shopping mall in city l_v that is new to u. Intuitively, a good recommender system should recommend v to u when he/she travels to l_v. However, the pure CF-based methods fail to do so. For the item-based CF [15, 21], there are few common users between v and v' according to the property of travel locality, resulting in the low similarity between the two items' user vectors. For the user-based CF [1], it is most likely that all the k nearest neighbors of user u live in the same city as u, and that few of them have visited v according to the property of travel locality.

To this end, we propose a location-content-aware recommender system (LCARS) that exploits both the location and content information of spatial items to alleviate the data sparsity problem, especially the *new city* problem. As is shown in Figure 1, LCARS consists of two main parts: offline modeling and online recommendation. The offline model, LCA-LDA, is designed to model user preferences to spatial items by simultaneously considering the following two factors in a unified manner. 1) *User Interest*: Music lovers may be more interested in concerts while Shopaholics would pay more attention to shopping malls. 2) *Local Preference*: When users visit a city, especially a city that is new to them, they are more likely to see local attractions and attend events that are popular in the city. Thus, the preferences of local people are a valuable resource for making a recommendation, especially when people travel to an unfamiliar area where they have little knowledge about the neighborhood. LCA-LDA can automatically learn both user interest and local preference from the user activity history. Exploiting local preference can address the issue of data sparsity to some extent. To further alleviate the data sparsity problem, LCA-LDA exploits the content information (e.g., item tags or category words) of spatial items to link content-similar spatial items together, facilitating people's travel not only near their home regions but also to cities that are new to them. It is worth mentioning that LCA-LDA can also capture the item co-occurrence patterns to link relevant items together, just like item-based collaborative filtering methods. To our best knowledge, ideas for unifying the influence of local preferences, collaborative filtering and content-based recommendation are unexplored and very challenging.

Given a querying user u with a querying city l_u, the online recommendation part computes a ranking score for each spatial item v within l_u by automatically combining u's interest and the local preference of l_u, which are learned offline by LCA-LDA. To speed up the process of online recommendation, we propose a scalable query processing technique for top-k recommendations which separates the offline scoring computation from online scoring computation to minimize the query time. Specifically, we partition all spatial items into locations at a given level such as cities. For each location, as is shown in Figure 1, we store K lists of spatial items that fall into the location and each list is sorted by the items' offline score in the corresponding dimension z. At query time, we retrieve all spatial items within l_u, and then extend the Threshold Algorithm (TA) to compute the top-k spatial items by combining the score of each candidate item from K scorers.

The primary contributions of our research are summarized as follows.

- We argue that local preference and item content information are important for modeling user preference and handling the data sparsity problem, and propose LCA-LDA, a novel location-content-aware probabilistic generative model that quantifies and incorporates both local preference and item content information in the spatial item recommendation process.
- We design a scalable query processing technique to improve the recommendation efficiency, enabling an online recommendation scenario.
- We conduct extensive experiments to evaluate the performance of our recommender system on two large-scale real data sets. The results show the superiority of our proposals in recommending spatial items for users, especially when traveling to new cities, in terms of both effectiveness and efficiency.

The remainder of the paper is organized as follows. Section 2 details the location-content-aware recommender system LCARS. We report the experimental results in Section 3. Section 4 reviews the related work and we conclude the paper in Section 5.

2. LOCATION-CONTENT-AWARE RECOMMENDER SYSTEM

In this section, we first introduce the key data structures and notations used in this paper, and then present the offline modeling part and online recommendation part of our proposed location-content-aware recommender system.

2.1 Preliminary

For ease of the following presentation, we define the key data structures and notations used in this paper. Table 1 lists the relevant notations used in this paper.

Definition 1. **(Spatial Item)** A spatial item v refers to either an event or venue generated in various EBSNs or LBSNs.

Definition 2. **(User Activity)** A user activity is a triple $(u,\ v,\ l_v)$ that means user u selects a spatial item v in location l_v. Information about the user activity history is given by $S \subseteq \mathcal{U} \times \mathcal{V} \times \mathcal{L}$, where user activities are positive observations in the past.

The dataset D used for our model learning consists of four elements, and they are users, spatial items, locations and content words, i.e., $(u,\ v,\ l_v,\ c_v) \in D$ where $u \in \mathcal{U}$, $v \in \mathcal{V}$, $l_v \in \mathcal{L}$, and $c_v \in \mathcal{C}_v$ (i.e., $\mathcal{C}_v$ denotes the content word set associated with spatial item v). Note that a spatial item may contain multiple content words. For an activity history record of a user u selecting a spatial item v in l_v, we have a set of four-tuples, i.e., $D_{uv} = \{(u,\ v,\ l_v,\ c_v) : c_v \in \mathcal{C}_v\}$.

Definition 3. **(User Profile)** For each user u in the dataset D, we create a user profile D_u, which is a set of four-tuples (i.e., $(u,\ v,\ l_v,\ c_v)$) associated with u. Clearly, $D_{uv} \subseteq D_u$.

SYMBOL	DESCRIPTION
N, V, M, C	the number of users, spatial items, locations, content words
$\mathcal{U}, \mathcal{V}, \mathcal{L}, \mathcal{C}$	the set of users, spatial items, locations, content words
$\mathcal{V}_l$	the set of spatial items located in location l
K	the number of topics
D_u	the profile of user u
v_{ui}	the spatial item of ith record in user profile D_u
θ_u	the interest of user u, expressed by a multinomial distribution over topics
θ'_l	the local preference of location l, expressed by a multinomial distribution over topics
ϕ_z	a multinomial distribution over spatial items specific to topic z
ϕ'_z	a multinomial distribution over content words specific to topic z
z_{ui}	the topic assigned to spatial item v_{ui}
l_{ui}	the location of spatial item v_{ui}
l_u	the querying location of the querying user u
c_{ui}	a content word describing spatial item v_{ui}
$\mathcal{C}_{ui}$	the set of content words describing spatial item v_{ui}
s_{ui}	if spatial item v_{ui} is generated by θ_u or $\theta'_{l_{ui}}$
β, β'	Dirichlet priors to multinomial distributions ϕ_z, ϕ'_z
α, α'	Dirichlet priors to multinomial distributions θ_u, θ'_l
λ_u	the mixing weight specific to user u; the parameter for sampling the binary variable s
γ, γ'	Beta priors to generate λ_u

Table 1: Notations used in the paper

Definition 4. **(Topic)** A topic z in a spatial item collection $\mathcal{V}$ is represented by a topic model ϕ_z, which is a probability distribution over spatial items, i.e., $\{P(v|\phi_z) : v \in \mathcal{V}\}$ or $\{\phi_{zv} : v \in \mathcal{V}\}$. By analogy, a topic in a content word collection $\mathcal{C}$ is represented by a topic model ϕ'_z, which is a probability distribution over content words, i.e., $\{P(c|\phi'_z) : c \in \mathcal{C}\}$ or $\{\phi'_{zc} : c \in \mathcal{C}\}$.

It is worth mentioning that each topic z corresponds to two topic models in our work, i.e., ϕ_z and ϕ'_z. This design enables ϕ_z and ϕ'_z to be mutually influenced and enhanced during the topic discovery process, facilitating the clustering of content-similar spatial items into the same topic with high probability.

Definition 5. **(User Interest)** The intrinsic interest of user u is represented by θ_u, a probability distribution over topics.

Definition 6. **(Local Preference)** The local preference of location l is represented by θ'_l, a probability distribution over topics. This modeling method can capture local folk-customs.

2.2 Offline Modeling

In this subsection, we first describe the offline modeling part of LCARS, a probabilistic generative model called LCA-LDA, and then present its inference process.

Model Description. The proposed offline modeling part, LCA-LDA, is a location-content-aware probabilistic mixture generative model that aims to mimic the process of human decision making on spatial items. This mixture model considers the user's personal interest and the influence of local preference in a unified manner, and automatically leverages the effect of the two factors. Specifically, given a querying user u with a querying city l_u, the likelihood that user u will prefer item v when traveling to city l_u, is sampled from the following LCA-LDA model.

$$P(v|\theta_u, \theta'_{l_u}, \phi, \phi') = \lambda_u P(v|\theta_u, \phi, \phi') + (1-\lambda_u) P(v|\theta'_{l_u}, \phi, \phi') \quad (1)$$

where $P(v|\theta_u, \phi, \phi')$ is the probability that spatial item v is generated according to the personal interest of user u, denoted as θ_u, and $P(v|\theta'_{l_u}, \phi, \phi')$ denotes the probability that spatial item v is generated according to the local preference of l_u, denoted as θ'_{l_u}. The parameter λ_u is the mixing weight which controls the motivation choice. That is, when deciding individual preference on v, user u is influenced by personal interest with probability λ_u, and is influenced by the local preference of l_u with probability $1 - \lambda_u$. It is worth mentioning that LCA-LDA holds personalized mixing weights for individual users, considering the differences between users in personality (e.g. openness, agreeableness).

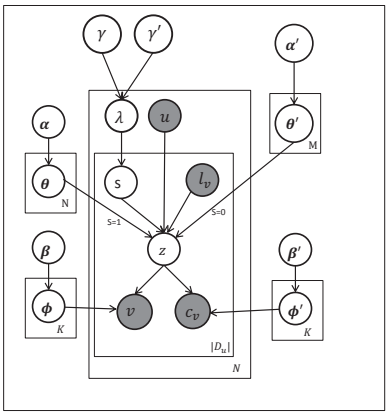


Figure 2: The Graphical Representation of LCA-LDA

To further alleviate the data sparsity problem, LCA-LDA incorporates the content information of spatial items. Thus, we reformulate Equation 1 as follows:

$$P(v|\theta_u, \theta'_{l_u}, \phi, \phi') = \sum_{c \in \mathcal{C}_v} P(v, c|\theta_u, \theta'_{l_u}, \phi, \phi') \quad (2)$$

$$P(v|\theta_u, \phi, \phi') = \sum_{c \in \mathcal{C}_v} P(v, c|\theta_u, \phi, \phi') \quad (3)$$

$$P(v|\theta'_{l_u}, \phi, \phi') = \sum_{c \in \mathcal{C}_v} P(v, c|\theta'_{l_u}, \phi, \phi') \quad (4)$$

where C_v is a set of content words describing spatial item v. In LCA-LDA, both user interest θ_u and local preference θ'_{l_u} are modeled by a multinomial distribution over latent topics. Each spatial item v is generated from a sample topic z. LCA-LDA also parameterizes a distribution over content words associated with each topic z, and thus topics are responsible for simultaneously generating both spatial items and their content words. Note that here we assume that items and their content words are independently conditioned on the topics. So, $P(v, c|\theta_u, \phi, \phi')$ and $P(v, c|\theta'_{l_u}, \phi, \phi')$ can be computed according to Equations 5 and 6. Parameter estimation in LCA-LDA is thus driven to discover topics that capture both item co-occurrence and content word co-occurrence patterns. This encodes our prior knowledge that spatial items having many common users or similar content should be clustered into the same topic with high probability. Figure 2 illustrates the generative process with a graph model. The generative process is also summarized in Algorithm 1 where $beta(.)$ is a Beta distribution, and γ and γ' are two parameters.

$$\begin{aligned} P(v, c|\theta_u, \phi, \phi') &= \sum_z P(v, c|z, \phi_z, \phi'_z) P(z|\theta_u) \\ &= \sum_z P(v|z, \phi_z) P(c|z, \phi'_z) P(z|\theta_u) \end{aligned} \quad (5)$$

$$\begin{aligned} P(v, c|\theta'_{l_u}, \phi, \phi') &= \sum_z P(v, c|z, \phi_z, \phi'_z) P(z|\theta'_{l_u}) \\ &= \sum_z P(v|z, \phi_z) P(c|z, \phi'_z) P(z|\theta'_{l_u}) \end{aligned} \quad (6)$$

Model Inference. Following the studies [26, 25], we use collapsed Gibbs sampling to obtain samples of the hidden variable assignment and to estimate unknown parameters $\{\theta, \theta', \phi, \phi', \lambda\}$

Algorithm 1: Probabilistic generative process in LCA-LDA

for *each topic* z **do**
  Draw $\phi_z \sim Dirichlet(\cdot|\beta)$;
  Draw $\phi'_z \sim Dirichlet(\cdot|\beta')$;
end
for *each* D_u *in* D **do**
  for *each record* $(u,\ v_{ui},\ l_{ui},\ c_{ui}) \in D_u$ **do**
    Toss a coin s_{ui} according to $bernoulli(s_{ui}) \sim beta(\gamma, \gamma')$;
    if $s_{ui} = 1$ **then**
      Draw $\theta_u \sim Dirichlet(\cdot|\alpha)$;
      Draw a topic $z_{ui} \sim multi(\theta_u)$ according to the interest of user u;
    end
    if $s_{ui} = 0$ **then**
      Draw $\theta'_{l_{ui}} \sim Dirichlet(\cdot|\alpha')$;
      Draw a topic $z_{ui} \sim multi(\theta'_{l_{ui}})$ according to the local preference of l_{ui};
    end
    Draw an item $v_{ui} \sim multi(\phi_{z_{ui}})$ from z_{ui}-specific spatial item distribution;
    Draw a content word $c_{ui} \sim multi(\phi'_{z_{ui}})$ from z_{ui}-specific content word distribution;
  end
end

in the LCA-LDA. As for the hyperparameters α, α', β, β', γ and γ', for simplicity, we take a fixed value (i.e., $\alpha = \alpha' = 50/K$, $\beta = \beta' = 0.01$, $\gamma = \gamma' = 0.5$). In the sampling procedure, we begin with the joint probability of all user profiles in the data set. Next, using the chain rule, we obtain the posterior probability of sampling topics for each four-tuple $(u,\ v,\ l_v,\ c_v)$. Specifically, we employ a two-step Gibbs sampling procedure. Due to space constraints, we show only the derived Gibbs sampling formulas, omitting the detailed derivation process. We first sample the coin s according to the posterior probability:

$$P(s_{ui} = 1|\boldsymbol{s}_{\neg ui}, \boldsymbol{z}, u, .) \propto \frac{n^{\neg ui}_{uz_{ui}} + \alpha_{z_{ui}}}{\sum_z (n^{\neg ui}_{uz} + \alpha_z)} \times \frac{n^{\neg ui}_{us_1} + \gamma}{n^{\neg ui}_{us_0} + n^{\neg ui}_{us_1} + \gamma + \gamma'} \tag{7}$$

$$P(s_{ui} = 0|\boldsymbol{s}_{\neg ui}, \boldsymbol{z}, u, .) \propto \frac{n^{\neg ui}_{l_{ui}z_{ui}} + \alpha'_{z_{ui}}}{\sum_z (n^{\neg ui}_{l_{ui}z} + \alpha'_z)} \times \frac{n^{\neg ui}_{us_0} + \gamma'}{n^{\neg ui}_{us_0} + n^{\neg ui}_{us_1} + \gamma + \gamma'} \tag{8}$$

where n_{us_1} is the number of times that $s = 1$ has been sampled in the user profile D_u; n_{us_0} is the number of times that $s = 0$ has been sampled in the user profile D_u; n_{uz} is the number of times that topic z has been sampled from the multinomial distribution specific to user u; n_{ux} is the number of times that latent region z has been sampled from the multinomial distribution specific to user u; the number $n^{\neg ui}$ with superscript $\neg ui$ denotes a quantity, excluding the current instance.

Then, we sample topic z according to the following posterior probability, when $s_{ui} = 1$:

$$P(z_{ui}|s_{ui} = 1, \boldsymbol{z}_{\neg ui}, \boldsymbol{v}, \boldsymbol{c}, u, .) \propto \frac{n^{\neg ui}_{uz_{ui}} + \alpha_{z_{ui}}}{\sum_z (n^{\neg ui}_{uz} + \alpha_z)} \frac{n^{\neg ui}_{z_{ui}v_{ui}} + \beta_{v_{ui}}}{\sum_v (n^{\neg ui}_{z_{ui}v} + \beta_v)} \frac{n^{\neg ui}_{z_{ui}c_{ui}} + \beta'_{c_{ui}}}{\sum_c (n^{\neg ui}_{z_{ui}c} + \beta'_c)} \tag{9}$$

when $s_{ui} = 0$:

$$P(z_{ui}|s_{ui} = 0, \boldsymbol{z}_{\neg ui}, \boldsymbol{v}, \boldsymbol{c}, u, .) \propto \frac{n^{\neg ui}_{l_{ui}z_{ui}} + \alpha'_{z_{ui}}}{\sum_z (n^{\neg ui}_{l_{ui}z} + \alpha'_z)} \frac{n^{\neg ui}_{z_{ui}v_{ui}} + \beta_{v_{ui}}}{\sum_v (n^{\neg ui}_{z_{ui}v} + \beta_v)} \frac{n^{\neg ui}_{z_{ui}c_{ui}} + \beta'_{c_{ui}}}{\sum_c (n^{\neg ui}_{z_{ui}c} + \beta'_c)} \tag{10}$$

where n_{zv} is the number of times that spatial item v has been generated by topic z, and n_{zc} denotes the number of times that content word c has been sampled from topic z.

After a sufficient number of sampling iterations, we can estimate the parameters θ, θ', ϕ, ϕ' and λ as follows:

$$\hat{\theta}_{uz} = \frac{n_{uz} + \alpha_z}{\sum_{z'} (n_{uz'} + \alpha_{z'})} \tag{11}$$

$$\hat{\theta}'_{lz} = \frac{n_{lz} + \alpha'_z}{\sum_{z'} (n_{lz'} + \alpha'_{z'})} \tag{12}$$

$$\hat{\phi}_{zv} = \frac{n_{zv} + \beta_v}{\sum_{v'} (n_{zv'} + \beta_{v'})} \tag{13}$$

$$\hat{\phi}'_{zc} = \frac{n_{zc} + \beta'_c}{\sum_{c'} (n_{zc'} + \beta'_{c'})} \tag{14}$$

$$\hat{\lambda}_u = \frac{n_{us_1} + \gamma}{n_{us_1} + n_{us_0} + \gamma + \gamma'} \tag{15}$$

2.3 Online Recommendation

A query in our recommendation task takes two arguments $(u,\ l_u)$: a querying user u with a querying city l_u to which u is going to travel. The result of a query is a ranked list of spatial items located at the querying city that match the querying user's preference. Once we have inferred LCA-LDA model parameters such as user interest θ, local preference θ', topics ϕ and ϕ', mixing weights λ, in the offline modeling phase, the online recommendation part computes a ranking score for each spatial item v within querying city l_u, and then returns top-k ranked spatial items as the recommendations.

In this part, we propose a ranking framework in Equation 16 which separates the offline scoring computation from the online scoring computation. Specifically, $F(l_u, v, z)$ represents the offline part of the scoring, denoting the score of spatial item v with respect to location l_u on dimension z that corresponds to topic z in the LCA-LDA model. Note that $F(l_u, v, z)$ is independent of querying users. The weight score $W(u, l_u, z)$ is computed in the online part, denoting the expected weight of the query $(u,\ l_u)$ on dimension z. It is worth mentioning that the main time-consuming components of $W(u, l_u, z)$ are also computed offline, and the online computation is just a simple combination process, as is shown in Equation 17. This design enables maximum precomputation for the problem considered, and in turn minimizes the query time. At query time, the offline scores $F(l_u, v, z)$ only need to be aggregated over K dimensions by a simple weighted sum function, in which the weight is $W(u, l_u, z)$. From Equations 17 and 18, we can see that $W(u, l_u, z)$ consists of two components, designed to model user interest and local preference respectively, and each component is associated with a kind of user motivation. $F(l_u, v, z)$ takes into account both the item co-occurrence information and the similarity of item contents to produce recommendations.

$$S(u, l_u, v) = \sum_z F(l_u, v, z) W(u, l_u, z) \tag{16}$$

$$W(u, l_u, z) = \hat{\lambda}_u \hat{\theta}_{uz} + (1 - \hat{\lambda}_u) \hat{\theta}'_{l_u z} \tag{17}$$

$$F(l_u, v, z) = \begin{cases} \hat{\phi}_{zv} \sum_{c_v \in \mathcal{C}_v} \hat{\phi}'_{zc_v} & v \in \mathcal{V}_{l_u} \\ 0 & v \notin \mathcal{V}_{l_u} \end{cases} \tag{18}$$

2.3.1 Threshold-Based Algorithm

For the online recommendation phase, we have to compute the preference scores of a querying user to all spatial items within the querying city and subsequently select the best k among them to recommend to the user. When the number of spatial items becomes larger (e.g., millions), computing the top-k spatial items for each query requires millions of vector operations. To speed up the online

process of producing recommendations, we extend the threshold-based algorithm [8] that is capable of correctly finding top-k results by examining the minimum number of spatial items.

We first partition all spatial items into locations at a predefined level such as cities. For each location, we precompute sorted lists of spatial items. This sorting is done offline according to $F(l, v, z)$ defined in Equation 18. Given K dimensions, we carry out this procedure for each dimension z (i.e., having spatial items on the same dimension in each sorted list). When receiving a query $q = (u, l_u)$, we first obtain K ranked list L_z, $z \in \{1, 2, ..., K\}$, of spatial items in location l_u. Algorithm 2 computes the top-k spatial items from these K lists and returns them in priority queue L. The algorithm maintains a priority queue PQ of the lists L_z where the priority of a list is the ranking score of the first spatial item in the list. This number is easily computed from the weight score $W(u, l_u, z)$ and the offline score $F(l_u, v, z)$ for the first spatial item v in the list. In each iteration, we pick the most promising spatial item from the list that is the head of PQ that we have not yet examined and add it to the results (L). The algorithm terminates early when the score of the k-th element of the results L is higher than the threshold ranking score which is computed in Algorithm 3.

3. EXPERIMENTS

In this section, we first describe the settings of experiments including the data sets, comparative approaches, and the evaluation method. We then report major experimental results on both the recommendation effectiveness and efficiency of our recommender system, followed by discussions.

3.1 Experimental Settings

3.1.1 Data sets

DoubanEvent is China's largest event-based social networking site where users can publish and participate in social events. On DoubanEvent, a social event is created by a user by specifying what, when and where the event is. Other users can express their intent to join events by replying online. This data set consists of 100,000 users, 300,000 events and 3,500,000 positive definite RSVPs. The following information is recorded when collecting the data: 1) user information, including user-id, user-name and user-home city; 2) event information, consisting of event-id, event-name, event-latitude, event-longitude, event-summary and its category; 3) user feedback information, including user-id and event-id. We make the dataset publicly available[1].

A publicly available LBSNs dataset, Foursquare [9], is also used in our experiment. It contains 11326 users and 1385223 check-ins. Note that this dataset does not contain item content information.

To utilize these two datasets in our proposed models, we preprocess them as follows: 1) we first employ Google Maps API[2] to partition all the spatial items into cities according to their latitudes and longitudes; 2) for the DoubanEvent dataset, we then use NLP toolkits[3] to extract a set of content words for each event from its summary and category description.

3.1.2 Comparative Approaches

We compare our proposed LCARS with the following six competitor methods, where the first four approaches are the existing recommender systems, and the last two recommender models correspond to the two main components of our proposed LCA-LDA.

- **User interest, social and geographical influences (USG)**: Following recent location-based recommendation work [31], a unified location recommendation framework is implemented which linearly fuses user interest, along with the social and geographical influences. The user interest component of USG is implemented by a traditional collaborative filtering technique, and the geographical influence is computed by a power-law probabilistic model that aims to capture the *geographical clustering phenomenon* that points of interest visited by the same user tend to be clustered geographically.

[1]http://net.pku.edu.cn/daim/yinhongzhi/index.html
[2]https://developers.google.com/maps/
[3]http://nlp.stanford.edu/software/index.shtml

Algorithm 2: Threshold-based algorithm

Input: A query $q = (u,\ l_u)$; ranked lists $(L_1, ..., L_K)$ for location l_u; inferred model parameters $\hat{\theta}_u$, $\hat{\theta}'_{l_u}$ and $\hat{\lambda}_u$;
Output: List L with all the k highest ranked spatial items according to query q;

{/*Priority queues (PQ and L) have five operations: $get()$ returns the head element from the queue; $remove()$ removes the head element from the queue; $get(k)$ returns the k-th element; $remove(k)$ removes the k-th element; $insert(element, priority)$ inserts an $element$ into the queue with a specific $priority$. */}
$PQ = \emptyset$;
$L = \emptyset$;
{/* lists L_z have three operations: $get()$ returns the head element from the list; $remove()$ removes the head element from the list; $hasMore()$ returns true if the list is non-empty. */}
$S_{Ta} = max$;
{/* variable S_{Ta} records the threshold score. */}
for $z = 1$ *to* K **do**
  $v = L_z.get()$;
  $PQ.insert(z, S(u, l_u, v))$;
end
$S_{Ta} = ComputeTA()$;
while $true$ **do**
  $nextListToCheck = PQ.get()$;
  $PQ.remove()$;
  $v = L_{nextListToCheck}.get()$;
  $L_{nextListToCheck}.remove()$;
  if $v \notin L$ **then**
    if $L.size() < k$ **then**
      $L.insert(v, S(u, l_u, v))$;
    end
    else
      $v' = L.get(k)$;
      if $S(u, l_u, v') > S_{Ta}$ **then**
        break;
      end
      if $S(u, l_u, v') < S(u, l_u, v)$ **then**
        $L.remove(k)$;
        $L.insert(v, S(u, l_u, v))$;
      end
    end
  end
  if $L_{nextListToCheck}.hasMore()$ **then**
    $v = L_{nextListToCheck}.get()$;
    $PQ.insert(nextListToCheck, S(u, l_u, v))$;
    $S_{Ta} = ComputeTA()$;
  end
  else
    break;
  end
end

Algorithm 3: Function ComputeTA()

Input: A priority queue PQ; A query $q = (u,\ l_u)$; ranked lists $(L_1, ..., L_K)$ for location l_u; inferred model parameters $\hat{\theta}_u$, $\hat{\theta}'_{l_u}$ and $\hat{\lambda}_u$;
Output: The threshold score S_{Ta};

$S_{Ta} = 0$;
for $i = 1$ *to* K **do**
  $z = PQ.get(i)$;
  $v = L_z.get()$;
  $S_{Ta} = S_{Ta} + W(u, l_u, z)F(l_u, v, z)$;
end

- **Category-based k-Nearest Neighbors Algorithm (CKNN)**: CKNN [3] projects a user's activity history into the category space and models user preference using a weighted category hierarchy. When receiving a query, CKNN retrieves all the users and items located in the querying city, formulates a user-item matrix online, and then applies a user-based CF method to predict the querying user rating of an unvisited item. Note that the similarity between two users in CKNN is computed according to their weights in the category hierarchy, making CKNN a hybrid recommendation method.
- **Item-based k-Nearest Neighbors Algorithm (IKNN)**: This method utilizes the user activity history to create a user-item matrix. When receiving a query, IKNN retrieves all users to find k nearest neighbors by computing the Cosine similarity between two users' item vectors. Finally, the spatial items in the user-specific querying city that have a relatively high ranking score will be recommended.
- **LDA**: Following previous works [11, 5], a standard LDA-based method is implemented as one of our baselines. Compared with our proposed LCA-LDA, this method neither considers the content information of spatial items, nor their location information. For online recommendation, the ranking score is computed using our ranking framework in Equation 16 where $F(l_u, v, z) = \hat{\phi}_{zv}$, $W(u, l_u, z) = \hat{\theta}_{uz}$.
- **Location-Aware LDA (LA-LDA)**: As a component of the proposed LCA-LDA model, LA-LDA means our method without considering the content information of spatial items. For online recommendation, the ranking score is computed using our proposed ranking framework in Equation 16 where $F(l_u, v, z) = \hat{\phi}_{zv}$ and $W(u, l_u, z) = \hat{\lambda}_u \hat{\theta}_{uz} + (1-\hat{\lambda}_u)\hat{\theta}'_{l_u z}$.
- **Content-Aware LDA (CA-LDA)**: As another component of the LCA-LDA model, CA-LDA means our method without exploiting the location information of spatial items, i.e., local preference. It can capture the prior knowledge that spatial items with the same or similar contents are more likely to belong to the same topic. This model is similar to the ACT model [26] in the methodology. For online recommendation, the ranking score is computed using our ranking framework in Equation 16 where $F(l_u, v, z) = \hat{\phi}_{zv} \sum_{c_v \in \mathcal{C}_v} \hat{\phi}'_{z c_v}$ and $W(u, l_u, z) = \hat{\theta}_{uz}$.

3.1.3 Evaluation methods

We evaluate both the effectiveness of the suggested recommendations and the efficiency for generating online recommendations.

Recommendation Effectiveness. To make an overall evaluation of the recommendation effectiveness of our proposed LCA-LDA, we first design the following two real settings: 1) querying cities are new cities to querying users; 2) querying cities are the home cities of querying users. We then divide a user's activity history into a test set and a training set. We adopt two different dividing strategies with respect to the two settings. For the first setting, we select all spatial items visited by the user in a non-home city as the test set and use the rest of the user's activity history in other cities as the training set. For the second setting, we randomly select 20% of spatial items visited by the user in personal home city as the test set, and use the rest of personal activity history as the training set.

According to the above designed dividing strategies, we split the user activity history S into the training data set $S_{training}$ and the test set S_{test}. To evaluate the recommender models, we adopt the testing methodology and the measurement Recall@k applied in [7, 5, 13, 32]. For each test case $(u,\ v,\ l_v)$ in S_{test}:

1. We randomly select 1000 additional spatial items located at l_v and unrated by user u. We assume that most of them will not be of interest to user u.
2. We compute the ranking score for the test item v as well as the additional 1000 spatial items.
3. We form a ranked list by ordering all the 1001 spatial items according to their ranking scores. Let p denote the rank of the test item v within this list. The best result corresponds to the case where v precedes all the random items (i.e., $p = 0$).
4. We form a top-k recommendation list by picking the k top ranked items from the list. If $p < k$ we have a hit (i.e., the test item v is recommended to the user). Otherwise we have a miss. The probability of a hit increases with the increasing value of k. When $k = 1001$ we always have a hit.

The computation of Recall@k proceeds as follows. We define hit@k for a single test case as either the value 1 if the test spatial item v appears in the top-k results, or else the value 0. The overall Recall@k are defined by averaging all test cases:

$$Recall@k = \frac{\#hit@k}{|S_{test}|} \tag{19}$$

where $\#hit@k$ denotes the number of hits in the test set, and $|S_{test}|$ is the number of all test cases.

Recommendation Efficiency. The efficiency of the online recommendation mainly depends on 1) the number of all spatial items in the user-specific querying city and 2) the number of spatial items recommended. Therefore, we test the efficiency of our proposed LCARS over these two factors.

3.2 Experimental Results

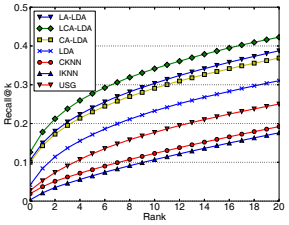
In this subsection, we first report the performance of our LCARS on the recommendation effectiveness and then compare the time costs of different recommendation algorithms.

3.2.1 Effectiveness of Recommendations

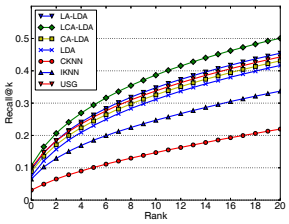
In this part, we first present the optimal performance with well-tuned parameters and then study the impact of model parameters.

Figure 3 reports the performance of the recommendation algorithms on DoubanEvent dataset. We show only the performance where k is in the range $[1...20]$, because a greater value of k is usually ignored for a typical top-k recommendation task. It is apparent that the algorithms have significant performance disparity in terms of top-k recall. As shown in Figure 3(a) where querying cities are new cities, the recall of LCA-LDA is about 0.33 when $k = 10$, and 0.42 when $k = 20$ (i.e., the model has a probability of 33% of placing an appealing event within the querying city in the top-10 and 42% of placing it in the top-20). Clearly, our proposed LCA-LDA model outperforms other competitor recommendation algorithms significantly. First, IKNN, CKNN and USG drop behind four other model-based methods, showing the advantage of using latent topic models to model users' preferences and produce recommendations. Second, LA-LDA outperforms LDA, justifying the benefit brought by considering local preferences. Third, CA-LDA exceeds LDA due to the advantages of taking item contents into consideration. Finally, LCA-LDA outperforms both LA-LDA and CA-LDA, showing the advantages of combining local preferences and item contents in a unified manner.

In Figure 3(b), we report the performance of all recommendation algorithms for the second setting where querying cities are home cities of querying users. From the figure, we can see that the trend of comparison result is similar to that presented in Figure 3(a). The

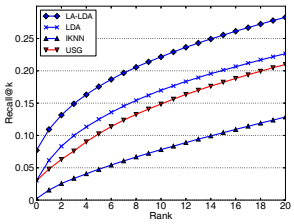


(a) Users Traveling in New Cities

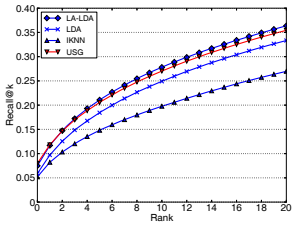


(b) Users Traveling in Home Cities

Figure 3: Top-k Performance on DoubanEvent

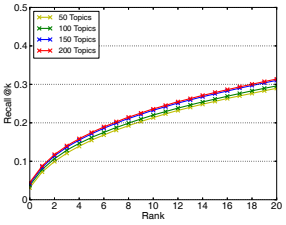


(a) Users Traveling in New Cities

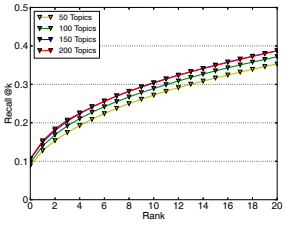


(b) Users Traveling in Home Cities

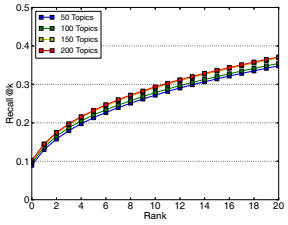
Figure 4: Top-k Performance on Foursquare



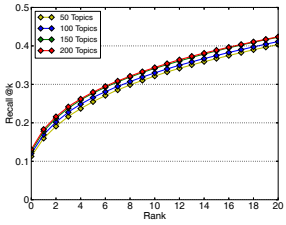
(a) LDA



(b) LA-LDA



(c) CA-LDA



(d) LCA-LDA

Figure 5: Impact of the Number of Latent Topics

main difference is that CKNN outperforms IKNN in Figure 3(a) while IKNN exceeds CKNN significantly in Figure 3(b), showing that the pure CF-based method (i.e., IKNN) better suits the setting where the user-item matrix is not very sparse, and the hybrid method (i.e., CKNN) is more capable of overcoming the difficulty of data sparsity, i.e., the *new city* problem. Another observation is that USG almost performs as well as LA-LDA, and outperforms LDA, CKNN and IKNN in the home city setting, verifying the benefit brought by considering the geographical influence. However, USG is still less effective than LCA-LDA in this setting. Furthermore, the performance of USG is poor in the new city setting, as shown in Figure 3(a), which shows that exploiting geographical influence cannot alleviate the *new city* problem since there is no activity history of the querying user in the new city.

Figure 4 reports the performance of the recommendation algorithms on the Foursquare dataset. We only compare LA-LDA, one component of our LCA-LDA model, with LDA, USG and IKNN since this dataset does not contain item content information. From the figure, we can see that the trend of comparison result is similar to that presented in Figure 3, and LA-LDA performs best, showing the advantage of exploiting the local preference.

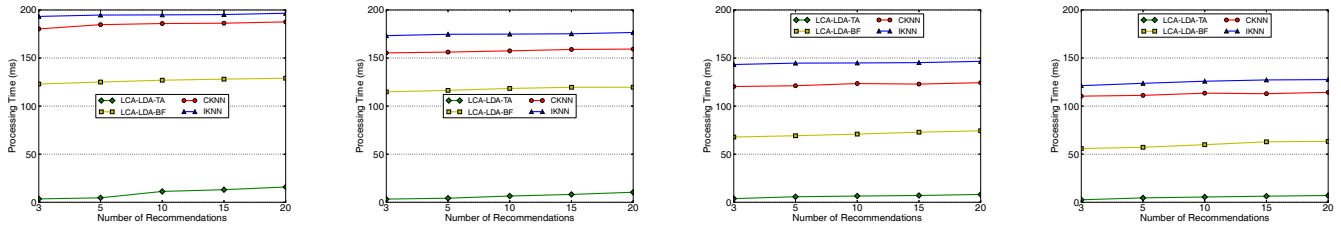
Impact of Model Parameters. Tuning model parameters, such as the number of topics for all topic models, is critical to the performance of models. We therefore also study the impact of model parameters on DoubanEvent dataset. Because of space limitations, we only show the experimental results for the new city setting.

As for the hyperparameters α, α', β, β', γ and γ', following existing works [26, 25], we empirically set fixed values (i.e., $\alpha = \alpha' = 50/K, \beta = \beta' = 0.01, \gamma = \gamma' = 0.5$). We tried different setups and found that the estimated topic models are not sensitive to the hyperparameters, but the performance of topic models such as LDA are slightly sensitive to the number of topics. Thus, we tested the performance of LDA, LA-LDA, CA-LDA and LCA-LDA models by varying the number of topics, and present the results in Figures 5(a) to 5(d). From the figures, we observe: 1) the Recall@k values of all latent topic-based recommender models slightly increase with the increasing number of topics; 2) the performance of latent topic-based recommender models does not change significantly when the number of topics is larger than 150; 3) LA-LDA, CA-LDA and LCA-LDA perform better than LDA under any number of topics, and LCA-LDA consistently performs best.

3.2.2 *Efficiency of Recommendations*

In the efficiency study on DoubanEvent, we tested 10000 querying users for the querying cities of Beijing, Shanghai, Guangzhou and Shenzhen respectively, by recommending a ranked list of events in each querying city for each test user. It is worth mentioning that there is different number of events in these four cities (i.e., $|\mathcal{V}_{Beijing}| > |\mathcal{V}_{Shanghai}| > |\mathcal{V}_{Guangzhou}| > |\mathcal{V}_{Shenzhen}|$). All the recommendation algorithms were implemented in Java (JDK 1.6) and run on a Linux Server with 32G RAM. For the online recommendation of LCARS, we adopt two methods to utilize the knowledge learnt offline by LCA-LDA to produce recommendations. The first is called LCA-LDA-TA proposed in Section 2.3.1, which extends TA algorithm to produce top-k recommendations. The second is called LCA-LDA-BF which uses a brute-force algorithm to produce top-k recommendations. In LCA-LDA-BF, we online compute the preference score of a test user to all events within the querying city and subsequently select the best k ones.

Figures 6(a)-6(d) present the average online efficiency of different methods, varying in the number of recommendations, for querying cities Beijing, Shanghai, Guangzhou, and Shenzhen respectively. For example, on average our proposed LCA-LDA-TA can find the top-10 event recommendations from about 72,000 events within Beijing in 11.4ms, from 51,780 events within Shanghai in 6.7ms, from 18,000 events within Guangzhou in 6.1 ms, and from 13,290 events within Shenzhen in about 5.4 ms. From the figures, we observe that 1) LCA-LDA-TA outperforms LCA-LDA-BF significantly in all querying cities, justifying the benefits brought by the TA algorithm; 2) both LCA-LDA-TA and LCA-LDA-BF are consistently better than CKNN and IKNN, showing that the model-based methods can produce faster responses to querying users than memory-based methods once the model parameters are learnt offline; 3) the time costs of all algorithms increase slowly with the increasing number of recommendations; 4) the time cost (TS) of each algorithm in four different cities can be ranked as follows: $TS_{Beijing} > TS_{Shanghai} > TS_{Guangzhou} > TS_{Shenzhen}$, showing



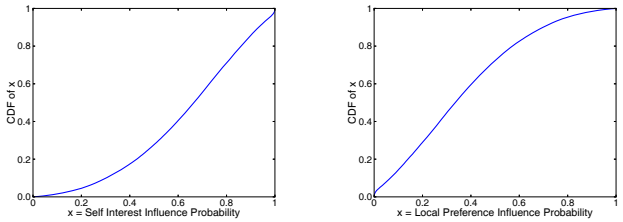
(a) Processing Time in Beijing (b) Processing Time in Shanghai (c) Processing Time in Guangzhou (d) Processing Time in Shenzhen

Figure 6: Efficiency w.r.t Recommendations

that a city with more events available requires more processing time to produce top-k recommendations within this city.

3.3 Local Preference Influence Study

In this section, we study the effects of personal interest and local preference on users' decision making. The self interest influence probability λ_u and the local preference influence probability $1-\lambda_u$ are learnt automatically in our proposed LCA-LDA model. Since different people have different mixing weights, we plot the distributions of both self interest and local preference influence probabilities among all users. The results on the DoubanEvent data set are shown in Figure 7, where Figure 7(a) plots the cumulative distribution of self interest influence probabilities, and Figure 7(b) shows the local preference influence probabilities. It can be observed that, in general, people's self interest influence is higher than the influence of the local preference. For example, Figure 7(a) shows that the self interest influence probability of more than 70% of users is higher than 0.5. The implication of this finding is that people mainly attend social events based on their self interests, and they sometimes attend popular local events regardless of their interests, especially when travelling in new cities. This finding also explains the superiority of LCA-LDA and LA-LDA in the recommendation performance (Section 3.2.1).



(a) Self Interest Influence (b) Local Preference Influence

Figure 7: Local Preference Influence Result (DoubanEvent)

4. RELATED WORK

In this section, we introduce related works, including general recommender systems and location-based recommendation.

4.1 General Recommender Systems

Collaborative filtering and content-based filtering techniques are two widely adopted approaches for recommender systems [1]. Both of them discover users' personal interests and utilize these discovered interests to find relevant items. Collaborative filtering techniques [5, 21] automatically suggest relevant items for a given user by referencing item rating information from other taste-similar users. The content-based recommendation [20] is based on the assumption that descriptive features of an item tell much about a user's preference for an item. Recommender systems using pure collaborative filtering approaches tend to fail when little knowledge about the user is known or when no one has interests similar to the user's. Although the content-based method is capable of coping with the lack of knowledge, it fails to account for community endorsement. As a result, a certain amount of research has focused on combining the advantages of both collaborative filtering and content-based methods [18, 4, 12]. Our proposal in this work is not only able to integrate the ideas behind collaborative filtering and content-based methods but also incorporates the influence of the local preference into the recommendation process.

4.2 Location-Based Recommendation

Regardless of the preference of an individual, some recent literatures [27, 33] focus on non-personalized spatial item recommendation systems which encapsulate public opinions on spatial items and provide users with the most interesting ones. Several studies [17, 2] address the problem of predicting the future locations of moving objects by using a model (e.g., a decision tree model or Hidden Markov Model) based on the mined trajectory patterns. Another branch of recent research focuses on learning user interest from the user's activity history to make personalized recommendations. Specifically, [14, 30, 24] deposited people's activity history into user-venue matrix where each row corresponds to a user's venue-visiting history and each column denotes a venue such as a restaurant. A user-based CF method is then employed to infer the user preference regarding an unvisited venue. Geo-measured friend-based collaborative filtering [30] produces recommendations by using only ratings from the querying user's social-network friends who live in the same city.

However, solely using a CF-based method, either the user-based or the item-based, cannot handle the data sparsity problem very well if we directly formulate user-venue matrix. Although literatures [19, 33, 5] applied latent factor models such as topic model and matrix factorization to a user-venue matrix to reduce the data sparsity to some extent, these methods do not work well in the *new city* setting because there are few overlapped users, possibly none, between spatial items which are located in home cities and new cities respectively. Instead of using traditional CF-based methods, [3] proposed a category-based similarity computation method which is able to find k-nearest neighbors for a querying user in a new city. Gao et al. [9] utilized the social network information to solve the "cold start" location prediction problem, with a geo-social correlation model to capture social correlations on LBSNs. Mao et al. [31] exploited the *geographical clustering phenomenon* to improve the recommendation performance, with a unified framework to linearly fuse both user interest and geographical influence. Woerndl et al. [29] developed a proactive context-aware model for mobile recommender systems which first analyzes the current situation and then computes the ranking scores of candidate items.

The interest in location-based data spans beyond the domain of spatial item recommendation (e.g., points of interest and events). Many recent literatures [23, 28, 6] have analyzed the interplay between users' mobility and their online social connections. Based on

the analysis results, they proposed a link prediction framework to recommend social connections based on users' physical mobility.

Our proposed location-content-aware recommender system distinguishes itself from the above-mentioned works in the following three aspects: 1) We project a user's activity history into a latent space which integrates the content knowledge of spatial items. Thus, we can recommend spatial items to a user in a new city by exploiting the content information about his/her preferred spatial items in other cities. 2) We take into account both user interest and local preference to produce recommendations. The local preference, which has previously been neglected, is a valuable resource for making a recommendation since people generally want to see local attractions and attend local popular events, especially when traveling to an unfamiliar city. 3) The idea of integrating local preference's influences, collaborative filtering and content-based methods into a probabilistic generative model is unexplored.

5. CONCLUSION

This paper proposed a location-content-aware recommender system, LCARS, which provides a user with spatial item recommendations within the querying city based on the individual interests and the local preferences mined from the user's activity history. LCARS can facilitate people's travel not only in their home area but also in a new city where they have no activity history. By taking advantage of both the content and location information of spatial items, our system overcomes the data sparsity problem in the original user-item matrix. We evaluated our system using extensive experiments based on two real data sets. According to the experimental results, our approach significantly outperforms existing recommendation methods in effectiveness. The results also justify each component proposed in our system, such as taking local preferences and item content information into account, and the proposed scalable query processing technique improves the efficiency of our approach significantly.

6. ACKNOWLEDGMENTS

This research is supported by the National Natural Science Foundation of China under Grant No. 60933004, 61073019 and 61272155. It is also partially supported by UTS Early Career Researcher Grant. This work is partially done when the first author visited SA Center for Big Data Research hosted in Renmin University of China. This Center is partially funded by a Chinese National "111" Project "Attracting International Talents in Data Engineering and Knowledge Engineering Research".

7. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.

[2] J. Alvarez-Lozano, J. A. García-Macías, and E. Chávez. User location forecasting at points of interest. In *LocalPeMA*, 2012.

[3] J. Bao, Y. Zheng, and M. F.Mokel. Location-based and preference-aware recomemnation using sparse geo-soical networking data. In *GIS*, 2012.

[4] J. Basilico and T. Hofmann. Unifying collaborative and content-based filtering. In *ICML*, 2004.

[5] W.-Y. Chen, J.-C. Chu, J. Luan, H. Bai, Y. Wang, and E. Y. Chang. Collaborative filtering for orkut communities: discovery of user latent behavior. In *WWW*, pages 681–690, 2009.

[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] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *RecSys*, pages 39–46, 2010.

[8] R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. In *PODS*, pages 102–113, 2001.

[9] H. Gao, J. Tang, and H. Liu. gscorr: modeling geo-social correlations for new check-ins on location-based social networks. In *CIKM*, pages 1582–1586, 2012.

[10] T. Horozov, N. Narasimhan, and V. Vasudevan. Using location for personalized poi recommendations in mobile environments. In *SAINT*, pages 124–129, 2006.

[11] X. Jin, Y. Zhou, and B. Mobasher. A maximum entropy web recommendation system: combining collaborative and content features. In *KDD*, pages 612–617, 2005.

[12] B. M. Kim, Q. Li, C. S. Park, S. G. Kim, and J. Y. Kim. A new approach for combining content-based and collaborative filters. *J. Intell. Inf. Syst.*, 27(1):79–91, 2006.

[13] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, pages 426–434, 2008.

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

[15] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

[16] 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.

[17] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. Wherenext: a location predictor on trajectory pattern mining. In *KDD*, pages 637–646, 2009.

[18] A. Popescul, D. M. Pennock, and S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *UAI*, pages 437–444, 2001.

[19] S. Rendle. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22, May 2012.

[20] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. *Recommender Systems Handbook*. Springer-Verlag New York, Inc., 2010.

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

[22] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo. Socio-spatial properties of online location-based social networks. In *ICWSM*, 2011.

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

[24] Y. Takeuchi and M. Sugimoto. Cityvoyager: an outdoor recommendation system based on user location history. In *UIC*, pages 625–636, 2006.

[25] J. Tang, S. Wu, J. Sun, and H. Su. Cross-domain collaboration recommendation. In *KDD*, pages 1285–1293, 2012.

[26] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: extraction and mining of academic social networks. In *KDD*, pages 990–998, 2008.

[27] P. Venetis, H. Gonzalez, C. S. Jensen, and A. Halevy. Hyper-local, directions-based ranking of places. *Proc. VLDB Endow.*, 4(5):290–301, 2011.

[28] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi. Human mobility, social ties, and link prediction. In *KDD*, pages 1100–1108, 2011.

[29] W. Woerndl, J. Huebner, R. Bader, and D. Gallego-Vico. A model for proactivity in mobile, context-aware recommender systems. In *RecSys*, pages 273–276, 2011.

[30] M. Ye, P. Yin, and W.-C. Lee. Location recommendation for location-based social networks. In *GIS*, pages 458–461, 2010.

[31] 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.

[32] H. Yin, B. Cui, J. Li, J. Yao, and C. Chen. Challenging the long tail recommendation. *Proc. VLDB Endow.*, 5(9):896–907, 2012.

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